89-4027

A NUMERICAL SOLUTION FOR THE DIFFUSION EQUATION IN HYDROGEOLOGIC SYSTEMS

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 89-4027

READEX FICHE best copy available

A NUMERICAL SOLUTION FOR THE DIFFUSION EQUATION IN HYDROGEOLOGIC SYSTEMS

By A. L. Ishii, R. W. Healy, and R. G. Striegl

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 89-4027

Urbana, Illinois

June 1989

DEPARTMENT OF THE INTERIOR

MANUEL LUJAN, JR., Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

District Chief
U.S. Geological Survey
4th Floor
102 East Main Street
Urbana, IL 61801

Copies of the report can be
purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Federal Center, Building 810
Box 25425
Denver, CO 80225

CONTENTS

	Page
Abstract	1
Introduction	1
Theoretical development	2
Molecular-diffusion equation	3
Analogous transport equations	4
Implementation of numerical model	7
Finite-difference approximation	7
Boundary and initial conditions	9
Solution by method of preconditioned conjugate gradients	10
Mass balance	13
Verification of numerical model	14
Molecular diffusion in one dimension	14
Heat conduction in two dimensions	16
Radial axisymmetric fluid flow with cylindrical coordinates	16
Application of numerical model	18
Grid design	19
Input parameters	20
Possible program modifications	21
Example application	21
Summary	54
References cited	55
Attachment A: Definition of Program Variables	58
Attachment B: Data Input Requirements	67
Attachment C: List of Program Routines	70
Attachment D: Program Listing	73

ILLUSTRATIONS

Figure 1.	One-dimensional diffusion system for an instantaneous time period, dt	4
2.	Arrangement of nodes for finite-difference approximation	8
3.	Comparison of numerical and analytical solutions for molecular diffusion in one dimension	15
4.	Comparison of numerical and analytical solutions for heat conduction in an anisotropic square plate	17
5.	Comparison of numerical and analytical solutions for radial fluid flow in a confined aquifer	18
6.	Geologic section to be modeled	23
7.	Example of a plot of the program output	53
8.	Simplified flow chart of major steps in program execution	71

TABLES

		Page
Table 1.	Analogous equations of flux	5
2.	Analogous transient flow equations	5
3.	Calculation of the effective diffusivities for the example application	24
4.	Input file for the example application	25
5.	Partial listing of output file for example application	31

CONVERSION FACTORS

For readers who prefer to use inch-pound units, conversion factors for the metric (International System) terms used in this report are listed below:

Multiply metric unit	By	To obtain inch-pound unit
meter (m)	3.281	foot (ft)
meter squared per second (m^2/s)	10.76	foot squared per second (ft^2/s)
meter squared per second (m^2/s)	1,549	inch squared per second (in^2/s)
meter squared per minute (m^2/min)	10.76	foot squared per minute (ft^2/min)
meter squared per day (m^2/d)	10.76	foot squared per day (ft^2/d)
pascal (Pa)	0.000009869 .0001450	atmosphere pound per square foot (lb/ft^2)
cubic meter per minute (m^3/min)	35.31	cubic foot per minute (ft^3/min)
grams per cubic meter (g/m^3)	0.06242	pound per cubic foot (lb/ft^3)

- -

Temperature may be converted from degrees Celsius (°C) to degrees Fahrenheit (°F) or from kelvins (K) to degrees Rankine (°R) as follows:

$$°F = 1.8 \times °C + 32$$
$$°R = 1.8 \times K$$

- -

Sea level: In this report, "sea level" refers to the National Geodetic Vertical Datum of 1929 (NGVD of 1929)--a geodetic datum derived from a general adjustment of the first-order level nets of both the United States and Canada, formerly called "Sea Level Datum of 1929."

SYMBOLS USED IN TEXT

Symbol	Dimension	Description
$[A]$	-	Coefficient matrix
$[\hat{A}]$	-	Conditioned coefficient matrix
b	L	Confined aquifer thickness
$\underline{b}$	-	Storage vector
$\underline{\hat{b}}$	-	Transformed storage vector
C	ML^{-3}	Concentration
$\underline{C}$	-	Concentration vector
$\underline{\overline{C}}$	-	Vector of concentration changes
C_A	ML^{-3}	Concentration of species A
C_0	ML^{-3}	Initial concentration
$\hat{C}p$	$L^2T^{-2}\theta^{-1}$	Specific heat at constant density
D	L^2T^{-1}	Molecular diffusivity
D_{AB}	L^2T^{-1}	Binary molecular diffusivity of species A into species B
D_{eff}	L^2T^{-1}	Effective molecular diffusivity
D_h	L^2T^{-1}	Hydraulic diffusivity
$[E]$	-	Coefficient matrix
$\underline{d}$	-	Vector constructed in the conjugate gradient algorithm
F	$ML^{-2}T^{-1}$	Mass flux in molecular diffusion
F_A	$ML^{-2}T^{-1}$	Mass flux of species A in molecular diffusion
g	-	Index denoting iteration
H	MT^{-3}	Heat flux
h	L	Total hydraulic head

SYMBOLS USED IN TEXT

Symbol	Dimension	Description
h_0	L	Initial value of total hydraulic head
h_E	$ML^{-1}T^{-2}$	Energy concentration
i	L^0	Index denoting location in horizontal (x or r) direction for finite-difference grid
I	-	Array index in a computer program
j	L^0	Index denoting location in vertical (z) direction for finite-difference grid
J	-	Array index in a computer program
k	-	Index denoting time step
k	$MLT^{-3}\theta^{-1}$	Thermal conductivity
K	-	Arbitrary constant
K	-	Element index in a computer program
K_{sat}	LT^{-1}	Saturated hydraulic conductivity
$[L \cdot L^T]$	-	Preconditioning matrix
[M]	-	Conditioned coefficient matrix
m	-	Ratio of anisotropy, lateral to vertical
n	-	Number of nodes in the finite-difference grid
NCOL	-	Number of columns in computer representation of model grid
NROW	-	Number of rows in computer representation of model grid
P	$ML^{-1}T^{-2}$	Pressure
P_0	$ML^{-1}T^{-2}$	Initial pressure

SYMBOLS USED IN TEXT

Symbol	Dimension	Description
$\underline{p}_g$	-	Vector constructed in the conjugate gradient algorithm, gth iteration
q	LT^{-1}	Specific fluid flux
Q	L^3T^{-1}	Volumetric flow rate
Q_E	MT^{-3}	Energy flux
r	L	Radius in x-direction
$\underline{r}$	-	Residual vector
$\hat{\underline{r}}$	-	Transformed residual vector
$[R]$	-	Residual matrix
ρ	ML^{-3}	Density
S	-	Storativity of a confined aquifer
$\underline{S}$	-	Storage vector
S_s	L^{-1}	Specific storage
t	T	Time
T	L^2T^{-1}	Transmissivity
u	-	Argument for the well function $= r^2/4D_ht$
x	L	Horizontal coordinate
$\underline{x}$	-	Concentration vector
$\hat{\underline{x}}$	-	Conditioned concentration vector
x_p	$ML^{-1}T^{-2}$	Partial pressure
z	L	Vertical coordinate
α	L^2T^{-1}	Thermal diffusivity
α_g	-	Conjugate gradient iteration parameter

SYMBOLS USED IN TEXT

Symbol	Dimension	Description
β_g	-	Conjugate gradient iteration parameter
ε	-	Arbitrary error criterion, tolerance at which iterations cease
θ	-	Temperature
η_D	-	Drained (air-filled) porosity
η_T	-	Total (air- and water-filled porosity)

A NUMERICAL SOLUTION FOR THE DIFFUSION EQUATION IN HYDROGEOLOGIC SYSTEMS

By A. L. Ishii, R. W. Healy, and R. G. Striegl

ABSTRACT

This report documents a computer code for the numerical solution of the linear diffusion equation in one or two dimensions in Cartesian or cylindrical coordinates. Applications of the program include molecular diffusion, heat conduction, and fluid flow in confined ground-water systems. The flow media may be anisotropic and heterogenous.

The model is formulated by replacing the continuous linear diffusion equation by discrete finite-difference approximations at each node in a block-centered grid. The resulting matrix equation is solved by the method of preconditioned conjugate gradients. The conjugate gradient method does not require the estimation of iteration parameters and is guaranteed convergent in the absence of rounding error. The matrices are preconditioned to decrease the steps to convergence. The model allows the specification of various boundary conditions for any number of stress periods and the output of a summary table for selected nodes showing flux and the concentration of the flux quantity for each time step. The model is written in a modular format for ease of modification.

The model is verified by comparison of numerical and analytical solutions for cases of molecular diffusion, two-dimensional heat transfer, and axisymmetric radial saturated fluid flow. Application of the model to a hypothetical two-dimensional field situation of gas diffusion in the unsaturated zone is demonstrated. The input and output files are included as a check on program installation. The definition of variables, input requirements, flow chart, and program listing are included in the attachments.

INTRODUCTION

Interest in the unsaturated zone as a pathway for the movement of liquids, gases, and solutes has increased in recent years. A recognition of the importance of the process of diffusion in both aeration and the transport of contaminant gases through the unsaturated zone has arisen. The diffusion equation describes this process in mathematical terms. An efficient algorithm, the method of preconditioned conjugate gradients, which is particularly suited for the solution of the diffusion equation, has also received increased attention. The method is not widely available as a matrix-solving algorithm in numerical models. This relatively new method is attractive in that it does not require the estimation of iteration parameters, as do other iterative methods, and has a comparatively good rate of convergence.

The report documents a numerical model that was developed to utilize this algorithm in a simple and efficient model that may be used in any process that is described by the diffusion equation. The model is a Fortran-77[1] com-puter program for the numerical solution of the linear diffusion equation in one or two dimensions in Cartesian or cylindrical coordinates. The model is formulated by replacing the continuous equation for a region with a set of finite-difference equations written for each node in a block-centered grid. The time derivatives are discretized by a backwards implicit approximation; the spatial derivative by a central difference. The resulting matrix equation is solved iteratively by the method of conjugate gradients.

The processes of heat transport and saturated fluid flow also are described by the diffusion equation. Because the fundamental laws are of identical form, it is possible to model heat conduction and saturated fluid flow using the same solution algorithms. This is accomplished by changing the constant of proportionality appropriately. The constant of proportionality is the same in all directions only if the medium is isotropic. In the case of an anisotropic medium in two dimensions, there are two principal directions of diffusivity. The principal directions may be aligned with the x- and z-axis to simplify the analytical and numerical solutions. In the model, the process modeled and the medium anisotropy are indicated by the values of the constant of proportionality and the ratio of anisotropy.

The derivation of the diffusion equation is explained in terms of molecular diffusion, and the analogous equations for heat conduction and saturated fluid flow are described. The model is verified by comparison with analytical solutions for simple cases of molecular diffusion, heat transfer, and axisymmetric fluid flow. The application of the model to a hypothetical field situation is demonstrated. The definition of variables, input requirements, flow chart, and program listing are included in the attachments.

THEORETICAL DEVELOPMENT

The diffusion equation is a parabolic partial differential equation that describes transient flow under concentration, temperature, heat, and hydraulic head gradients. This equation may be derived from the fundamental laws that describe the steady-state flux of each quantity by using the law of conservation of mass or energy to write the equations of continuity. For simplicity, derivation of the equation is presented for the case of molecular diffusion in one dimension. The extension to the analogous transport equations will be described in the next section.

---

[1]Use of brand and trade names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

Molecular-Diffusion Equation

Ordinary gaseous diffusion is the movement of gas resulting solely from the concentration gradient of the gas in an isobaric system. Fick's first law states that the mass flux of the diffusing gas is proportional to its change in concentration over space. This may be expressed for a binary system in one dimension as

$$F_A = - D_{AB} \frac{dC_A}{dx} ,$$

where F_A is the mass flux of the diffusing gas A, $ML^{-2}T^{-1}$;
D_{AB} is the constant of proportionality commonly known as the binary molecular diffusivity for gas A into gas B, L^2T^{-1};
C_A is the concentration of the diffusing gas A, ML^{-3}; and
x is distance in the direction of decreasing C_A, L.

The value of D_{AB} depends on physical constants of the gas species as well as the temperature and pressure of the system. Values of D_{AB} for various gases may be found in a standard reference such as the Perry's Chemical Engineers' Handbook (Perry and others, 1984).

The diffusion equation is written by considering conservation of mass. Figure 1 shows an element in a one-dimensional closed system. Since there is no mass flux into or out of the system, the net mass flux into any element must equal the time rate of change of storage (concentration) within the element. Thus, the equation of continuity for gas A is

$$\frac{\partial F_A}{\partial x} = \frac{\partial}{\partial x} \left(D_{AB} \frac{\partial C_A}{\partial x} \right) = \frac{\partial C_A}{\partial t} .$$

Since the total density, pressure, and temperature of the system are constant in this analysis, D_{AB} is constant and the equation of continuity may be expressed as

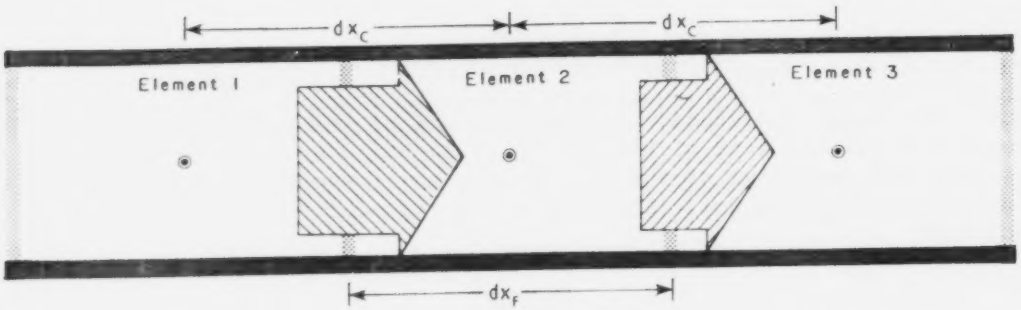
$$D_{AB} \frac{\partial^2 C_A}{\partial x^2} = \frac{\partial C_A}{\partial t} ,$$

which is termed Fick's second law.

For two and three dimensions, C_A is differentiated twice over each additional direction. For cylindrical coordinates, Fick's second law is

$$D_{AB} \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial C_A}{\partial r} \right) + \frac{\partial^2 C_A}{\partial z^2} \right] = \frac{\partial C_A}{\partial t} .$$

The derivation of the diffusion equation is discussed in depth in Bird and others (1960, p. 554-560).

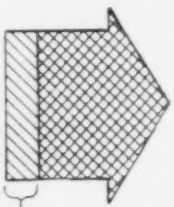


Flux in $= D\frac{dC}{dx_c}$

where D is the diffusivity between elements 1 and 2, and $\frac{dC}{dx_c}$ is the concentration gradient from element 1 to element 2.

Flux out $= D\frac{dC}{dx_c}$

where D is the diffusivity between elements 2 and 3, and $\frac{dC}{dx_c}$ is the concentration gradient from element 2 to element 3.



Change in flux for element 2 $= \frac{dF}{dx_f} = \frac{dC}{dt}$

For the instantaneous time period, dt, the change in flux is the change in storage, dC, for element 2.

Figure 1.--One-dimensional diffusion system for an instantaneous time period, dt.

Analogous Transport Equations

The fundamental laws for heat transfer and saturated fluid flow are analogous to Fick's first law. Fourier's law of heat conduction states that heat flux is proportional to the temperature gradient. Darcy's law states that specific flux of a fluid is proportional to the total head gradient. These relations and the constants of proportionality are summarized in table 1.

The equations describing unsteady heat conduction and incompressible saturated fluid flow are analogous to Fick's second law. The equations are presented in table 2.

Table 1.--Analogous equations of flux

Name of law/ physical process	Flux	Variable gradient	Constant of proportionality	Fundamental law
Fick's First/ molecular diffusion	Mass flux, F, $ML^{-2}T^{-1}$	Concentration, C, ML^{-3}	Molecular diffusivity, D, L^2T^{-1}	$F = -D \frac{dC}{dx}$
Fourier's/ heat conduction	Heat flux, H, MT^{-3}	Temperature, θ	Thermal conductivity, k, $MLT^{-3}\theta^{-1}$	$H = -k \frac{d\theta}{dx}$
Fourier's/ heat transfer	Energy flux, Q_E, MT^{-3}	Energy concentration, h_E, $ML^{-1}T^{-2}$	Thermal diffusivity, α, L^2T^{-1}	$Q_E = -\alpha \frac{dh_E}{dx}$
Darcy's/ saturated fluid flow	Specific fluid flux, q, LT^{-1}	Total hydraulic head, h, L	Saturated hydraulic conductivity, K_{sat}, LT^{-1}	$q = K_{sat} \frac{dh}{dx}$

Table 2.--Analogous transient flow equations

Physical process	Transient flow equation
Molecular diffusion	$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2}$
Heat conduction	$\frac{\partial \theta}{\partial t} = k \frac{\partial^2 \theta}{\partial x^2}$
Heat transfer	$\frac{\partial h_E}{\partial t} = \alpha \frac{\partial^2 h_E}{\partial x^2}$
Saturated fluid flow	$\frac{\partial h}{\partial t} = K_{sat} \frac{\partial^2 h}{\partial x^2}$

The equations presented in tables 1 and 2 are in a simple one-dimensional form. These equations are linear and, thus, valid only for the range in which the constants of proportionality do not depend on the value of the dependent variable (concentration, temperature, or hydraulic head). The constants of proportionality depend upon the properties of the flux quantity and the flow media and are hereby discussed separately for the various phenomena.

For molecular diffusion, the constant of proportionality, D, depends on the diffusing species and the medium into which it is diffusing. The binary gaseous diffusion coefficient, D_{AB}, is modified for diffusion in a porous media.

The conceptual reason for the difference is that the average path length is increased for each gas molecule as it travels around solid particles. The constant of proportionality, D_{AB}, may be multiplied by a tortuosity factor (less than 1.0) to account for this effect. Investigators have found that the factor is related to the volume of voids in the medium (drained porosity in soils) (Cunningham and Williams, 1980). In addition, gas molecules may be adsorbed onto the solids. There are various modifications to the diffusion coefficient reported in the literature to account for these and other effects (Weeks and others, 1982).

The flow of heat is described in table 1 in two equivalent ways. In the classical statement of Fourier's law, the flux quantity is temperature. The constant of proportionality is the thermal conductivity, k.

The equation described as heat transfer has energy concentration as the flux quantity. The constant of proportionality is thermal diffusivity, α. Thermal diffusivity is related to thermal conductivity by the equation

$$\alpha = k/\rho\hat{C}_p,$$

where ρ is the density of the medium, ML^{-3};

$\hat{C}p$ is the specific heat capacity at constant pressure, $L^2T^{-2}\theta^{-1}$; and

$\rho\hat{C}_p\theta$ may be thought of as the thermal energy concentration per unit volume, h_E (Bird and others, 1960, p. 503).

In a saturated porous medium, the diffusion of energy is the time rate of change of the energy stored in the solid matrix and the liquid in response to the temperature gradient. In an unsaturated system, it is the time rate of change of the energy stored in the solid matrix and the gas in response to the temperature gradient. Unlike the cases of gas diffusion and fluid flow, the solid matrix may contribute positively to the transfer of the energy rather than strictly impeding it. An average thermal conductivity (diffusivity) may be calculated for the system by adding the conductivities (diffusivities) for the solid matrix and liquid or gas fractions weighted by the volume of each (Voss, 1984, p. 35-37).

For saturated fluid flow in confined porous media, the specific storage, S_s, may be important. The specific storage is a function of the volume of voids in the medium, the compressibility of the liquid and the medium solids, and the arrangement of the solids. The saturated hydraulic conductivity, K_{sat}, is divided by the storage term, S_s, to produce the hydraulic diffusivity, D_h (Freeze and Cherry, 1979), which is the effective constant of proportionality for the model. In the field of water-well hydraulics, analytical solutions utilize the confined-aquifer parameters--transmissivity, T, and storativity, S. These parameters are defined for a confined aquifer as

$$T = K_{sat} b, \ L^2T^{-1},$$

and

$$S = S_s \ b, \text{ dimensionless,}$$

where b is the aquifer thickness, L. Hydraulic diffusivity is then expressed as

$$D_h = T/S = K_{sat}/S_s, \ L^2T^{-1}.$$

Discussion of these parameters is found in Freeze and Cherry (1979, p. 58-62).

IMPLEMENTATION OF NUMERICAL MODEL

The numerical model was formulated by representing the domain using a block-centered grid. The spatial derivatives in two dimensions are approximated by the five-point, finite-difference, linear operator written for each node. The harmonic mean is used to compute the internodal values for all variables. Use of the harmonic mean insures continuity across cell boundaries for variable grids and accurate no-flow boundaries (Trescott and others, 1976). The set of equations are solved by the method of preconditioned conjugate gradients. Accuracy of the solution may be checked by referring to the mass balance.

Finite-Difference Approximation

The following finite-difference approximation is described for molecular diffusion. The form of these equations is identical to those describing heat conduction and saturated fluid flow, since the equations describing those processes are analogous.

The two-dimensional differential equation that describes the change in concentration of a diffusing gas species at each point in a two-dimensional system with respect to time is approximated for each node i,j (see fig. 2) at time k, by the following implicit finite-difference equation:

$$(\Delta x_i \Delta z_j)\ \frac{C_{i,j,k} - C_{i,j,k-1}}{t_k - t_{k-1}} = \left(\frac{D}{\Delta x}\right)_{i+\frac{1}{2},j} (C_{i+1,j,k} - C_{i,j,k})\Delta z_j$$

$$+ \left(\frac{D}{\Delta x}\right)_{i-\frac{1}{2},j} (C_{i-1,j,k} - C_{i,j,k})\Delta z_j + \left(\frac{D}{\Delta z}\right)_{i,j+\frac{1}{2}} (C_{i,j+1,k} - C_{i,j,k})\Delta x_i$$

$$+ \left(\frac{D}{\Delta z}\right)_{i,j-\frac{1}{2}} (C_{i,j-1,k} - C_{i,j,k})\Delta x_i ,$$

where i,j are subscripts indicating node location;

k is a subscript indicating time step;

are subscripts indicating the harmonic mean between adjacent nodes where the harmonic mean approximation for $\left(\frac{D}{\Delta x}\right)_{i+\frac{1}{2},j}$ between nodes i and i+1 is $\frac{2D_{i,j}D_{i+1,j}}{\Delta x_i D_{i+1,j} + \Delta x_{i+1} D_{i,j}}$;

C is the concentration of the diffusing gas, ML^{-3};

D is the effective diffusion coefficient for the diffusing gas, L^2T^{-1};

t is time;

Δx is the length of the finite-difference cell in the x-direction, L; and

Δz is the length of the finite-difference cell in the z-direction, L.

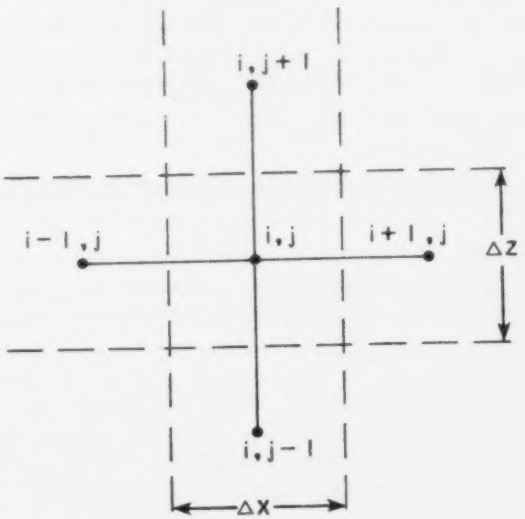


Figure 2.--Arrangement of nodes for finite-difference approximation.

The diffusion equation for the axisymmetric case may be approximated in cylindrical coordinates at each point (r,z) by the following implicit finite-difference equation for each node i,j at time k:

$$2\pi r\Delta r_i \Delta z_j \left(\frac{C_{i,j,k} - C_{i,j,k-1}}{t_k - t_{k-1}}\right) = \left(\frac{D}{\Delta r}\right)_{i+\frac{1}{2},j} (C_{i+1,j,k} - C_{i,j,k})\Delta z_j 2\pi \left(\frac{r+\Delta r_i}{2}\right)$$
$$+ \left(\frac{D}{\Delta r}\right)_{i-\frac{1}{2},j} (C_{i-1,j,k} - C_{i,j,k})\Delta z_j 2\pi \left(\frac{r-\Delta r_i}{2}\right)$$
$$+ \left(\frac{D}{\Delta z}\right)_{i,j+\frac{1}{2}} (C_{i,j+1,k} - C_{i,j,k}) 2\pi r \Delta r_i$$
$$+ \left(\frac{D}{\Delta z}\right)_{i,j-\frac{1}{2}} (C_{i,j-1,k} - C_{i,j,k}) 2\pi r \Delta r_i .$$

The resulting set of n equations in n unknowns, where n is the total number of nodes, may be written in matrix form (Lapidus and Pinder, 1982) as

$$[E]\ \underline{\bar{C}}^g = \underline{S}^{g-1} - [E]\ \underline{C}^{g-1},$$

where [E] is a sparse symmetric pentadiagonal coefficient matrix of n^2 elements;

g is a superscript indicating iteration;

$\underline{C}^{g-1}$ is the concentration vector;

$\underline{\bar{C}}^g$ is the vector of concentration changes, $C^g - C^{g-1}$; and

$\underline{S}^{g-1}$ is the solved storage vector.

The set of equations are solved by the iterative method of preconditioned conjugate gradients.

Boundary and Initial Conditions

In order to obtain a unique solution to the diffusion equation, it is necessary to specify the boundary conditions (Ritger and Rose, 1968). Furthermore, since diffusion and conduction processes depend upon gradients in the time dependent variable, the initial conditions are also required (Mercer and Faust, 1980). Two types of boundaries may be specified in the model: Dirichlet and Neumann.

For a Dirichlet-type node, the dependent variable has a specified value at a boundary. For the ordinary gaseous diffusion case discussed earlier, this condition is

$$C = K,$$

where K is any constant; therefore,

$$\frac{dC}{dt} = 0.$$

This condition is known as constant concentration, temperature, or head depending on the flux quantity. It also is referred to as a source or sink, depending on whether its value is higher or lower than the local concentration. Examples of conditions in hydrogeologic systems modeled with Dirichlet-type nodes include the atmospheric source of oxygen at constant concentration, a radioactive pluton, which is a continuous source of heat at a constant temperature, or a constant-head river boundary to a ground-water system.

For the Neumann-type node, the change in the dependent variable normal to the boundary is specified. Thus, the derivative of the gradient function is specified. This is expressed as

$$D \frac{\partial C}{\partial x} = K$$

at some x along the boundary. This condition is known as specified flux. It also is referred to as continuous recharge or discharge depending on whether flux is positive or negative. When K = 0, the boundary is called a no-flow or impermeable boundary. Examples of conditions in hydrogeologic systems modeled with Neumann-type nodes include the continuous injection of a gas tracer, an insulating boundary, and continuous pumping of a well.

The program is initialized with a no-flow boundary around the model grid. This is equivalent to an impermeable zone around the region to be modeled. Concentrations at all nodes must be set to an initial value. Flux is initialized at zero for the entire grid.

The simulation period is divided into stress periods during which the boundary conditions are constant. Each stress period is divided into time steps. Early in a simulation, when gradients are large, transient flow conditions typically change considerably. The rate of change diminishes rapidly after the first few time steps. For efficiency, the size of each succeeding time step may be increased by a constant factor up to a predetermined size. In general, truncation error is proportional to the size of the time step; consequently, efficiency and accuracy need to be balanced in setting the time-step parameters (Remson and others, 1971).

Solution by Method of Preconditioned Conjugate Gradients

The method of conjugate gradients is an iterative technique that may be efficiently applied to large, sparse matrices such as those arising from the discretization of parabolic partial differential equations (Wong, 1979). The diffusion equation is of this form. Iterative techniques require less storage than direct methods for sparse matrices since the zero diagonals between the main diagonal and the outermost non-zero diagonal do not need to be stored. The accuracy of the solution is determined, in part, by the size of the convergence criterion and the maximum number of iterations allowed. The preconditioning technique transforms the matrix equation, and then the generalized conjugate gradient method is applied to the transformed matrix.

The method of conjugate gradients has additional properties that make it desirable for this application. The estimation of iteration parameters (the maximum and minimum eigenvalues of the coefficient matrix) is not required for convergence. Convergence is guaranteed in a maximum of n steps (where n is the number of nodes in the model matrix) and residual error decreases monotonically. This means that in the absence of round-off error the difference between succeeding iterative solutions decreases with each iteration. The method takes advantage of the distribution of the eigenvalues (internal as well as the maximum and minimum) so that convergence ordinarily occurs in far fewer than n steps (Concus and others, 1976, p. 317). The rate of convergence compares favorably with the Chebyshev iteration (Wong, 1979, p. 967), the Strongly Implicit Procedure (SIP), and the Successive Over-Relaxation (SOR) methods (Kuiper, 1984, p. 10).

The matrix equation derived earlier (omitting the iteration superscripts) is

$$[E]\,\underline{\overline{C}} = \underline{S} - [E]\,\underline{C}.$$

Reordering this into a standard mathematical form of this equation, from which the iterated residual may be derived, gives

$$[A]\underline{x} = \underline{b},$$

where $[A] = [E]$, $\underline{x} = \underline{C}$, and $\underline{b} = \underline{S}$.

The preconditioning matrix $[L \cdot L^T]$ is selected so that the condition number of the product, [M], of $[L \cdot L^T]^{-1}[A]$ is less than that of [A], and the eigenvalues are clustered in groups. The condition number is a measure of the ratio of the relative error to the relative residual. If the condition number is large, the relative error may be much larger than the relative residual. These factors decrease the theoretical steps to convergence (Wong, 1979).

The equation to be solved is now transformed to

$$[\hat{A}]\underline{\hat{x}} = \underline{\hat{b}},$$

where $[\hat{A}] = [L \cdot L^T]^{-1}\,[A]$,

$\underline{\hat{x}} = \underline{x}$, and

$\underline{\hat{b}} = [L \cdot L^T]^{-1}\underline{b}$.

The conjugate gradient iterations are then performed on the transformed residual $\underline{\hat{r}} = \underline{\hat{b}} - [\hat{A}]\underline{\hat{x}}$.

The method used to precondition the coefficient matrix was an incomplete decomposition of the matrix into a sparse upper and lower triangular matrices [L] and $[L^T]$ known as Incomplete Cholesky factorization (Meijerink and van der Vorst, 1977). The non-zeros of the coefficient matrix [A] are equated to those of the preconditioning matrix $[L \cdot L^T]$ and non-zeros coefficients in $[L \cdot L^T]$ not present in [A] are ignored, so that $[L \cdot L^T]$ equals [A]+[R] where [R] is small.

The conjugate gradient algorithm computes the iteration parameter d by minimizing the function $(\underline{x}-\hat{\underline{x}})^T$ [A] $(\underline{x}-\hat{\underline{x}})$ where $\underline{x}$ is the exact solution of $[A]\underline{x} = \underline{b}$ and $\hat{\underline{x}}$ is the best approximate solution along the line $\hat{\underline{x}} = \underline{x}_g + \alpha_g$ where $\underline{p}_g$ is determined recursively to be A-conjugate ($\underline{p}_g^T[A]\underline{p}_g = 0$). Since the true solution is not known, it is necessary to remove $\underline{x}$ by computing $\underline{r}_g = \underline{b} - [A]\underline{x}_g = [A](\underline{x}-\underline{x}_g)$. The known vector $\underline{b}$ and $[A]x_{g+1}$ may be omitted by computing $\underline{r}_{g+1}$ recursively as $\underline{r}_{g+1} = \underline{r}_g - \alpha_g [A]\underline{p}_g$.

The parameter α_g is

$$\alpha_g = \frac{\underline{d}_g^T \underline{r}_g}{\underline{p}_g^T[A]\underline{p}_g}$$

where $\underline{d}_g = [M]^{-1}\underline{r}_g$.

To construct the set of A-conjugate vectors each $\underline{p}_g$ must be orthogonal to $\underline{p}_{g-1}$ by using the equation

$$\underline{p}_g = \underline{d}_g + \beta_g \underline{p}_{g-1}$$

where $\beta_g = \dfrac{\underline{d}_g^T \underline{r}_g}{\underline{d}_{g-1}^T \underline{r}_{g-1}}$.

The conjugate gradient algorithm involves iterating over the following equations:

$$\underline{d}_g = [M]^{-1}\underline{r}_g;$$

$$\underline{p}_g = \underline{d}_g, \quad g = 0;$$

$$\beta_g = \frac{\underline{d}_g^T \underline{r}_g}{\underline{d}_{g-1}^T \underline{r}_{g-1}} ;$$

$$\underline{p}_g = \underline{d}_g + \beta_g \underline{p}_{g-1};$$

$$\alpha_g = \frac{\underline{d}_g^T \underline{r}_g}{\underline{p}_g^T[A]\underline{p}_g} ;$$

$$\underline{x}_{g+1} = \underline{x}_g + \alpha_g \underline{p}_g;$$

$$\underline{r}_{g+1} = \underline{r}_g - \alpha_g [A]\underline{p}_g.$$

The iterations are stopped when the maximum absolute value of C is less than an input error criterion, ε, for all nodes. If this occurs before the input maximum number of iterations has been reached, then the model is assumed to have converged upon a solution. This is equivalent to saying that the magnitude of improvement per iteration has fallen below a minimum level or

$$\alpha_g \rho_g < \varepsilon,$$

where ε is selected by considering machine accuracy, computation costs, and the problem requirements.

Mass Balance

The numerical accuracy and precision of the solution is checked by computing a mass balance for each time step. Convergence to within the desired tolerance does not guarantee that the solution is accurate for the physical process due to round-off errors and the residuals resulting from the discretization of continuous functions.

The principle of the conservation of mass (and energy) requires that the net flux must equal the change in storage for a closed system. The difference between net flux and net storage is called the mass residual.

The net flux is determined by summing the fluxes through constant flux and concentration nodes. Flux out of a node is considered positive, and flux into a node is negative. The change in storage is computed by subtracting the initial storage from the final storage for a time step for all nodes.

A measure of relative error can be computed by dividing the mass residual by the change in storage plus initial concentration (Konikow and Bredehoeft, 1978, p. 14). Relative error is equal to

$$\Delta t \Sigma F + \Delta t \Sigma \left[\left(\frac{D}{\Delta z} \right)_{i,j+\frac{1}{2}} \Delta x_i (C_{i,j} - C_{i,j+1}) + \left(\frac{D}{\Delta z} \right)_{i,j-\frac{1}{2}} \Delta x_i (C_{i,j} - C_{i,j-1}) \right.$$

constant flux nodes — constant concentration nodes

$$\left. + \left(\frac{D}{\Delta x} \right)_{i-\frac{1}{2},j} \Delta z_j (C_{i,j} - C_{i+1,j}) + \left(\frac{D}{\Delta x} \right)_{i+1,j} \Delta z_j (C_{i,j} - C_{i-1,j}) \right]$$

$$- \sum_{i,j} (C_{i,j,k} - C_{i,j,k-1}) \Delta x_i \Delta z_j \Big/ \sum_{i,j} \left[C_{i,j,o} - (C_{i,j,k} - C_{i,j,k-1}) \right] \Delta x_i \, \Delta z_j$$

where F is the fixed flux value for each node.

VERIFICATION OF NUMERICAL MODEL

The performance of the model was evaluated by comparing the numerical model solution with analytical solutions for three verification tests--molecular diffusion in one dimension, heat conduction in two dimensions, and radial fluid flow using the cylindrical coordinate system option. The solutions include those of Crank (1956) for molecular diffusion, Carslaw and Jaeger (1959) for heat conduction, and Theis (1935) for saturated fluid flow. Mass balance is reported for each verification test.

Molecular Diffusion in One Dimension

The first verification test is for molecular diffusion in one dimension. Molecular diffusion is described for the transient case by Fick's second law, given in the molecular diffusion equation section as

$$\frac{\partial C_A}{\partial t} = D_{AB} \frac{\partial^2 C_A}{\partial x^2} .$$

The initial condition for the problem is zero concentration everywhere. The boundary conditions are

$$C = C_0 \text{ at } x = 0.$$

and

$$C = 0. \text{ at } x = \infty$$

at all times. One solution (Crank, 1956) is

$$C_A\,(x,t) = C_0 \operatorname{erfc}(x/2\sqrt{D_{AB}t}),$$

where x is the distance, in meters;

t is time, in seconds;

C_A is concentration of diffusing species A, in grams per cubic meter;

C_0 is concentration of species A at $x = 0$, in grams per cubic meter;

D_{AB} is the coefficient of diffusion for species A into species B, in meters squared per day; and

erfc is the complementary error function.

For this example, the concentrations have been converted to a partial pressure in pascals (Pa) using the standard atmospheric temperature and pressure at sea level (STP). Partial pressures may be converted to concentration in grams per cubic meter by using the following formula:

$$\left(\frac{Xp}{101{,}325\ \text{Pa}}\right) 41.56 \left(\frac{\text{moles}}{\text{m}^3}\right)_{STP} \frac{MW}{\text{mole}} = C,$$

where Xp is the partial pressure, in pascals;

C is the concentration, in grams per cubic meter; and

MW is the mass of a mole of gas molecules, in grams (molecular weight).

The following values were assumed:

$$X_p = 10.0 \text{ Pa at } x = 0$$

and

$$D_{AB} = 1.244 \text{ meters squared per day } (m^2/d).$$

The 20-meter grid was spaced uniformly at 0.5 m in the x-direction. The initial time step size was 0.10 second and was increased by a factor of 1.5 up to a maximum of 150 seconds for the first day. The time step was then increased to 0.10 day (8,640 seconds) and increased by a factor of 1.5 to a maximum of 10 days.

The numerical and analytical solutions for molecular diffusion were nearly identical (fig. 3). Mass balance error was small, ranging from 0 to 0.46 percent.

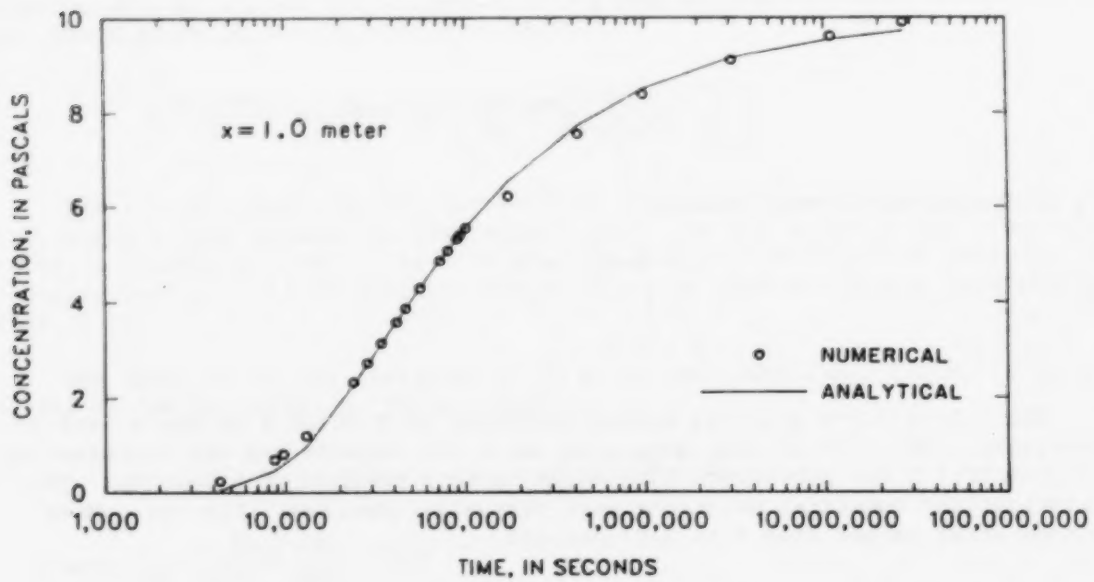


Figure 3.--Comparison of numerical and analytical solutions for molecular diffusion in one dimension.

Heat Conduction in Two Dimensions

The second verification test is for two-dimensional anisotropic heat flow in a thin square plate. The phenomenon is described for the transient case by the two-dimensional anisotropic form of the diffusion equation presented in table 2:

$$\frac{\partial \theta}{\partial t} = k \frac{\partial^2 \theta}{\partial x^2} + mk \frac{\partial^2 \theta}{\partial z^2},$$

where m is the ratio of thermal conductivity in the z direction to that in the x direction (anisotropy ratio), k_z/k_x, dimensionless. The initial condition for the problem is a temperature (θ) of 0.0 °C everywhere in the plane at time (t) equals zero. The boundary conditions are

$$\theta(x,z,t) = 1.0 \text{ for } t > 0.0 \text{ at}$$

$$x = 1.0,\ 0 \leq y \leq 1.0;$$

$$z = 1.0,\ 0 \leq x \leq 1.0;$$

which is equivalent to saying that the left and bottom edges of the plate are constant sources of heat. The temperature at any point in the plate at any time is given by the following analytical solution (Carslaw and Jaeger, 1959, p. 173):

$$\theta(x,y,t) = 1 - \frac{16}{\pi^2}\left(\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} e^{-k(2n+1)^2\pi^2 t/4} \cos \frac{(2n+1)\pi x}{2}\right)$$

$$\left(\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)} e^{-mk(2n+1)^2\pi^2 t/4} \cos \frac{(2n+1)\pi z}{2}\right).$$

The following values were assumed:

$$k = 0.001 \text{ m}^2/\text{s}$$

and

$$m = 4.0.$$

The 1.0- x 1.0-m grid was spaced uniformly at 0.05128 m in the x- and z-directions. The initial time step size was 0.001 seconds and was increased by a factor of 1.5 for subsequent time steps up to a maximum of 5 seconds. The analytical and numerical solutions were virtually identical (fig. 4). Mass balance error ranged from 0 to 3.07 percent.

Radial Axisymmetric Fluid Flow with Cylindrical Coordinates

The third verification test is for a pumping well in a confined aquifer using the cylindrical coordinate system option. The hydraulic parameters of

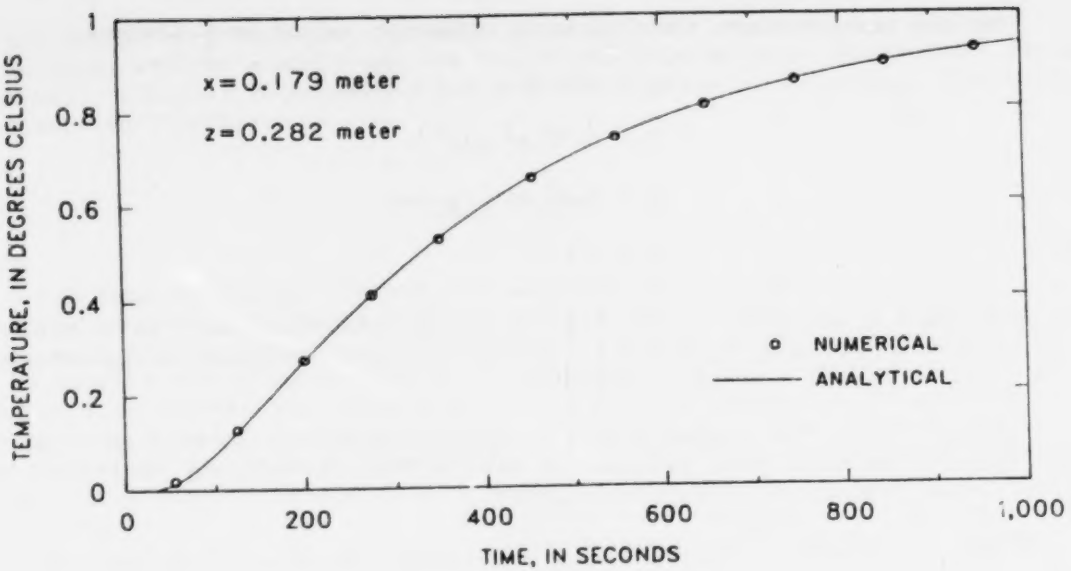


Figure 4.--Comparison of numerical and analytical solutions for heat conduction in an anisotropic square plate.

interest are usually transmissivity, T, and storativity, S. These are combined to yield the hydraulic diffusivity, $D_h = T/S$. Ground water flow to a pumping well in a confined axisymmetric aquifer may be described in cylindrical coordinates by the following equation:

$$\frac{\partial h}{\partial t} = D_h\left(\frac{1}{r}\frac{\partial h}{\partial r} + \frac{\partial^2 h}{\partial r^2}\right).$$

The initial condition for the problem is that the head is equal to h_0 everywhere in the aquifer at time equals zero. At the start of the simulation, water is withdrawn from the aquifer at a constant volumetric flow rate, Q. The boundary condition at an infinite radial distance from the origin is a constant head of h_0.

The head, h, at any distance, r, from the well and time, $t > o$, is given by the following solution (Theis, 1935):

$$h(r,t) = h_0 - \frac{Q}{4\pi T}\int_u^\infty \frac{e^{-u}}{u}\,du,$$

where $u = \frac{r^2 S}{4Tt} = \frac{r^2}{4Dt}$, and

the integral on the right hand side of the equation is known as the well function, W(u). Values for W(u) are tabulated in Lohman (1972, p. 16).

For the test problem, the following parameter values were assumed:

$$h_0 = 100 \text{ m};$$

$$T = 0.20 \text{ m}^2 \text{ min}^{-1};$$

$$Q = 5 \text{ m}^3 \text{ min}^{-1}; \text{ and}$$

$$S = 2 \times 10^{-5}.$$

The grid spacing was made variable in the radial direction, with an initial size of 0.05 m increased by a factor of 1.5 for each subsequent increment, up to a total of 66,665 m in 68 increments.

Figure 5 shows the numerical and analytical solutions plotted as total head versus time at a fixed distance of 2.69 m from the origin. Relative error ranged from 4.4×10^{-2} to 1.11 percent.

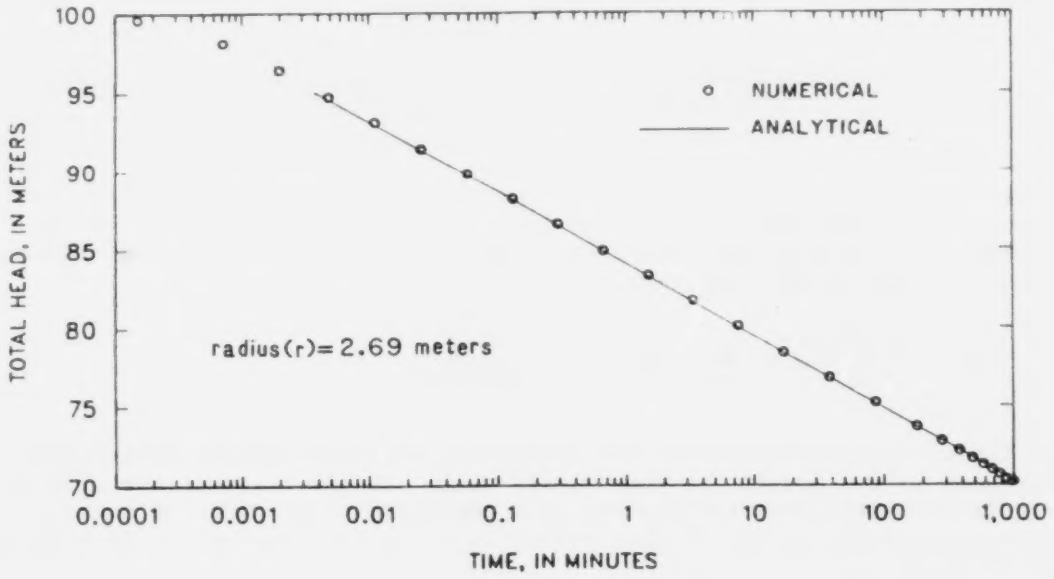


Figure 5.--Comparison of numerical and analytical solutions for radial fluid flow in a confined aquifer.

APPLICATION OF NUMERICAL MODEL

The model may be used to represent a one-dimensional flow, a two-dimensional vertical or horizontal flow, or an axisymmetric one- or two-dimensional flow. In this section the design of the block-centered grid, the choice of boundary and initial conditions, and time-step and iteration parameters will be discussed. The dimensions of parameters used and the modularity

of the diffusivity routines also will be described. These topics will be illustrated with an example application of the program to a hypothetical field problem. Technical requirements and possible program modifications also will be described briefly.

Grid Design

The simplest design for the grid elements is to consider the block-centered uniform grid overlaid on the section of interest with the nodes designated according to their locations. In a hydrogeologic system, the area to be modeled may be subdivided into regions having different diffusivity or conductivity properties. Each node is assigned the index for the parameter set that describes the flow properties in the medium in the block around it. The properties are assumed constant for the block; thus, an extremely heterogeneous area will require many more nodes to describe the parameter distributions adequately. If the area to be modeled is anisotropic, the x- and z-axes must be aligned with the principal directions of the diffusivity or conductivity tensor.

It is desirable to have the nodes coincide with relevant features (for example, wells, impermeable aquifer layers, trench boundaries, or heat sources). In order to achieve this, it may be necessary to refine the mesh by decreasing the spacing between nodes (Δx and Δz) or implement a variable spacing scheme; however, in designing the grid it should be remembered that truncation error is roughly proportional to the grid spacing. It has been suggested that the ratio of spacing of adjacent nodes should not exceed 1.5 to avoid large truncation errors (Trescott and others, 1976, p. 30). It is important that areas with large gradients be adequately covered with sufficient nodes to describe the changes. A distinct contrasting zone of diffusivity or conductivity should not be represented by a single row or column of nodes since the program computes an interblock diffusivity (or conductivity) which is an average value of adjacent nodes. The node spacing should be sufficiently small to have at least two nodes for any distinct contrasting zone included. More may be necessary if such a zone induces large gradients. Computation time and memory storage requirements are related to the number of nodes used; hence, efficiency and accuracy must be balanced in the grid design. Sensitivity analyses and consideration of mass balance results are required for each separate application. All physical boundaries that influence the region of interest should be included. If it is impractical to include a distant boundary, its effect should be simulated by making model boundaries sufficiently large to avoid artificial boundary effects. This should be verified by sensitivity analyses. Impermeable boundaries do not have to be explicitly entered since the model grid is initialized with a no-flow boundary around the region, unless the boundary is irregular.

The boundary conditions may be imposed for the entire simulation period or as a step function for selected stress periods. The initial conditions may be set to represent the steady-state condition if desired, however, the simulation will respond to these conditions unless the system is in stable equilibrium at the start of the simulation.

Although the grid is input as a two-dimensional array, it is changed internally to one-dimensional array or vector of NROW by NCOL elements, where NROW is the number of rows in the grid and NCOL is the number of columns in the grid. Vector storage reduces the computational time because computers find elements in a vector more quickly than in a two-dimensional array. Elements are assigned so that if node I,J is element K

node I-1,J is K-1,

node I,J-1 is K-NCOL,

node I,J+1 is K+NCOL, and

node I+1,J is K+1.

The vector identification of element I,J is the product of (I-1) and NCOL+J.

Input Parameters

The units of the input parameters may be in any system convenient to the user; however, the selected units must be consistent for all model input. Confusion may arise due to the several possible ways of describing the flux quantity. For example, mass concentration may be in mass per volume, parts per million, or partial pressures. Thermal gradients may be described in terms of temperature or energy concentration. The constants of proportionality usually have the dimensions L^2T^{-1} except for thermal conductivity, which has the dimensions $MLT^{-3}\theta^{-1}$. The units used to express the specified constant-flux nodes must be consistent with the units of the constants of proportionality and with the units of length and time used in setting up the model grid.

The simulation parameters include the time-related variables such as the initial time-step size, time-step multiplier, maximum time-step size, stress-period length, and number of stress periods. These variables will differ according to the problem being simulated, consequently, it is not possible to state a single rule for selecting values for these variables. The time steps should be small enough so that the concentration or head does not change significantly in a single time step. The determination of a significant change is up to the user's judgment; however, a value of about 10 percent has been suggested (Voss, 1984). Concentration, temperature, or head changes are expected to be more rapid at the start of a stress period; hence, the time steps may be increased by some factor as the simulation progresses. The value of the maximum time step is problem dependent and the user's judgment must be applied to select an appropriate value of the rapidness of change after the start of a stress period. The appropriateness of the time-step selection may be checked by numerical experimentation involving comparisons of results from simulations with various time steps.

The iteration parameters are the maximum number of iterations and the error criterion for closure. As the error criterion is decreased, the number of iterations required will increase. If convergence is not obtained within some limit of iterations, the temporal or spatial discretization may be too

coarse or the error criterion too small for the specified number of iterations. If the concentration values are oscillating as the maximum number of iterations are reached, the problem may be the temporal or spatial discretization.

Possible Program Modifications

The numerical algorithms are coded in Fortran-77 in the structured programming style. Each function within the program is in a modular subroutine called from the main program or program subunits. All input and output is handled in the main program except that which relates to the boundary conditions for individual stress period which are read in a subroutine called at specified time intervals. The subroutine and function descriptions and program flow chart are in attachment C.

The constants of proportionality are read in as material parameters. The constants are stored in a function, which is called as needed throughout the program. A function for storativity with a default value of 1.0 also is provided. The constant of proportionality may be decoupled into parameters unique to the flux quantity and the medium with appropriate modification of the two function modules. As an example, the effective diffusivity for gas diffusion in a porous medium may be decoupled into separate diffusivity and storativity functions. Within the diffusivity function, one parameter may describe the diffusion of gas into air (binary diffusion coefficient) and others describe the impeding effect of the solid particles. Parameters read into the storativity routine might include the gas-solid-liquid distribution coefficients to describe storage by dissolution and sorption. The basic equations may be modified to include reaction terms linked to concentration, if required, to account for radioactive decay or biological consumption. An example of such modification to the diffusion equation is found in Weeks and others (1982, p. 1370). The program is dimensioned to accept up to 8 parameters for up to 20 different materials.

The program has been compiled and run on the PR1ME 750 computer with the F77 Rev. 20.2.2 compiler, the SUN 3/260 computer with the SUN release 3.4 F77 compiler both with and without the floating point accelerator, and the IBM AT computer with the Ryan-McFarland Version 2.11 compiler. The Ryan-McFarland compiled version required 174 kilo bytes for storage of the executable program as dimensioned in this report. A desirable modification is changing the dimensions of arrays to fit individual program needs in order to reduce memory requirements. The amount of memory required to run the program is dependent upon the size of the model grid, the convergence criterion selected, the simulation time period, and the number of constant head or flux nodes specified.

Example Application

This section describes the application of the model to a hypothetical field situation. The input and output files are included so that the user may test the model on a different computer system. The simulation was run on a PR1ME 750 computer with the F77 revision 20.2.2 compiler. Model output is

plotted using Integrated Software Systems Corporation DISSPLA version 10.0 software plotting package. The user should not expect a plot of the results to match the illustrated plot exactly because different interpolation schemes are employed for different software, and all are sensitive to the levels of precision available on the operating system.

The area to be modeled is a geologic section of unsaturated glacial deposits (fig. 6). Methane is being produced from wastes in a trench (left boundary) and is diffusing into the unsaturated zone. The problem is to determine the concentration of methane, as a partial pressure, at the site 15 years after burial. It is assumed that the concentration of methane in the trench has a mean constant value of 3.0 Pa during the period simulated. This partial pressure is sufficiently low to exclude convection by total pressure differentials, since total atmospheric pressure is approximately 100,000 Pa.

The diffusivity of methane in free air at STP (standard temperature, 273.2 K, standard pressure, 1 atmosphere) is reported in Katz and others (1959, p. 100) as 1.693 m^2/d. It is corrected by the equation of Bird and others (1960, p. 507) for ambient pressure and temperature conditions. The equation is

$$D_{AB} = D_{AB}(1.0 \text{ atm}/P)(\theta/273.2 \text{ K})^{1.823},$$

where D_{AB} is the diffusion constant at STP, in meters squared per day;

P is the mean annual pressure at the site, atmosphere; and

θ is the mean annual temperature at the site, K.

The mean annual temperature is assumed to be 10.3 °C in the air and 12.8 °C below the surface. The pressure at the site is assumed to be 0.973 atmospheres for the zone from the surface to 15 m below the surface. These represent typical continental conditions.

The effective diffusivity for gas diffusion in the soil is determined by the equation of Millington and Quirk (1960) relating effective diffusivity to the drained (air-filled) and total (air- and water-filled) porosities. The equation is

$$D_{eff} = D_{AB} \frac{\eta_D^{10/3}}{\eta_T^2},$$

where D_{eff} is the effective diffusivity coefficient, in meters squared per day;

η_D is the drained porosity, dimensionless; and

η_T is the total porosity, dimensionless.

Table 3 shows the drained and total porosities, the diffusivity, and the effective diffusivity for each layer. The effective diffusivity was input to the model as the constant of proportionality (table 4, data set 13). The ratio of anisotropy is one since each layer is assumed isotropic.

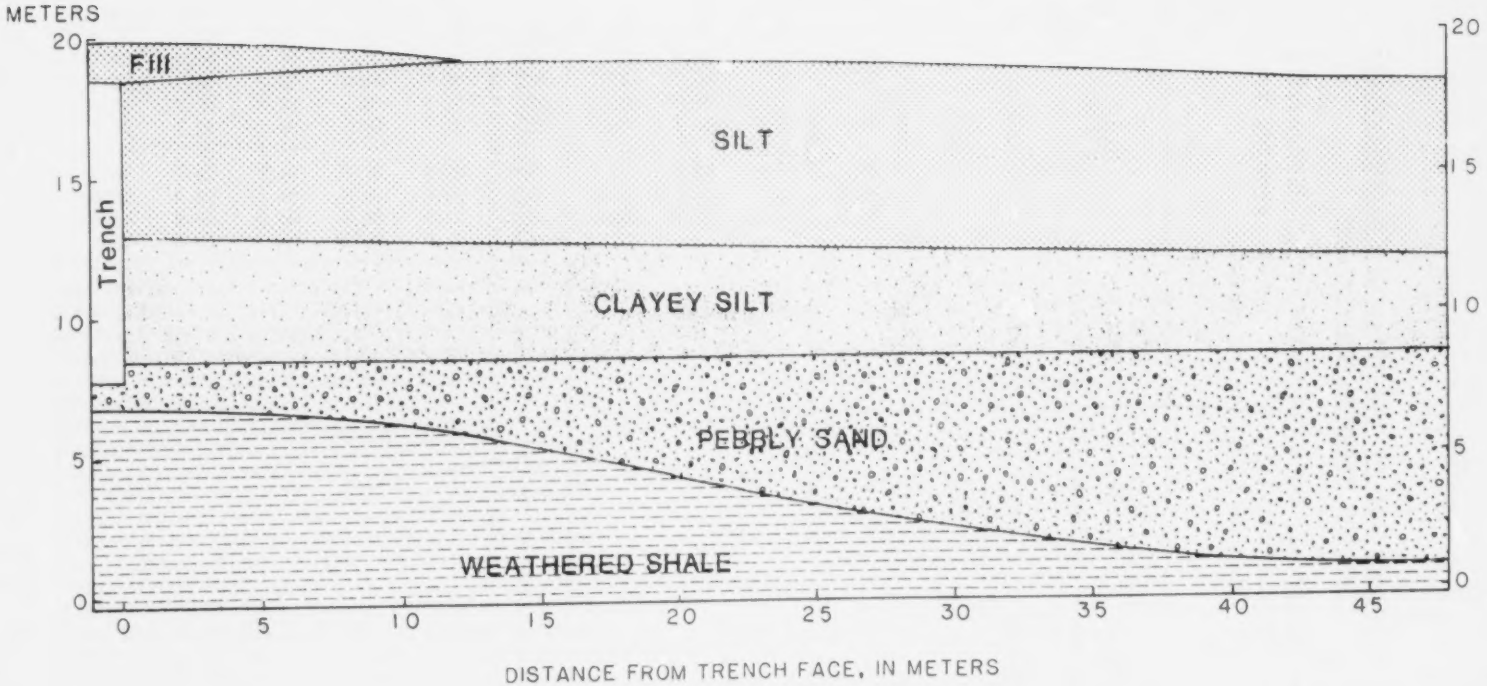


Figure 6.--Geologic section to be modeled.

Table 3.--Calculation of the effective diffusivities for the example application

[η_D, drained porosity; η_T, total porosity, D_{AB}, diffusivity in air; D_{eff}, effective diffusivity; m^2d^{-1}, meters squared per day; dashes indicate no data]

Material identifier	Material description	η_D (dimensionless)	η_D (dimensionless)	D_{AB} (m^2d^{-1})	D_{eff} (m^2d^{-1})
1	atmosphere	--	--	1.693	1.862
2	silt	.30	.43	1.892	.1850
3	clayey silt	.02	.32	1.892	.00004012
4	pebbly sand	.29	.35	1.892	.2493
5	weathered shale bedrock, partially saturated[1]	0	.03	--	.0000017

[1]For the weathered saturated shale bedrock, a value arbitrarily close to zero was selected to reflect a nearly impermeable boundary.

Table 4.--Input file for example application

```
                                                                                                                    DATA SET        VARIABLES
123456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789                              1          HEADER
      1        53        36         5         2       100         0         0         0                                   2          NPER,NCOL,NROW,NMAT,NPROP,MAXIT,NPR,RAD,NMB
   5600      0.10       1.5       75.      .001                                                                         3          TIMEMAX, DELT, CDELT, DELTMAX, EPS
      0       1.0        .0       1.0                                                                                  4          IX, FACTX, IZ, FACTZ
   1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30                        5          DELX
   1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30
   1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30
   1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30
   1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30      1.30
   1.30      1.30      1.30
   0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60                        6          DELZ
   0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60
   0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60      0.60
   0.60      0.60      0.60      0.60      0.60      0.60
1111111111111111111111111111111111111111111111111111111111                                                         8          MAT(I)
2222222211111111111111111111111111111111111111111111111111
2222222222211111111111111111111111111111111111111111111111
2222222222222222222222222221111111111111111111111111111111
1112222222222222222222222222222222222221111111111111111111
1112222222222222222222222222222222222222211111111111111111
1112222222222222222222222222222222222222221111111111111111
1112222222222222222222222222222222222222222211111111111111
1112222222222222222222222222222222222222222222111111111111
1112222222222222222222222222222222222222222222211111111111
1112222222222222222222222222222222222222222222222111111111
1112222222222222222222222222222222222222222222222222111111
1112222222222222222222222222222222222222222222222222222222
1113333333333222222222222222222222222222222222222222222222
1113333333333333333333333333333332222222222222222222222222
1113333333333333333333333333333333333333333333333333333333
1113333333333333333333333333333333333333333333333333333333
1113333333333333333333333333333333333333333333333333333333
1113333333333333333333333333333333333333333333333333333333
1113333333333333333333333333333333333333333333333333333333
1114444444444444444444444444444444444444444444444444444444
4444444444444444444444444444444444444444444444444444444444
4444444444444444444444444444444444444444444444444444444444
5555555554444444444444444444444444444444444444444444444444
5555555555554444444444444444444444444444444444444444444444
5555555555555544444444444444444444444444444444444444444444
5555555555555554444444444444444444444444444444444444444444
5555555555555555444444444444444444444444444444444444444444
5555555555555555544444444444444444444444444444444444444444
5555555555555555554444444444444444444444444444444444444444
5555555555555555555444444444444444444444444444444444444444
```

Table 4.--Input file for example application--Continued

```
5555555555555555555555555555544444444444444444444444
5555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555
     3     .18
  5600        1      172
     1       1.862          1.0                                  9        IC, FACTC
     2        .1850         1.0                                 12        PLENGTH, NB1, NB2
     3    .00004012         1.0                                 13        K, PARAM
     4        .2493         1.0
     5    .0000017          1.0
     5           2        1    3.00                             14        M, N, NT, C1
     5           3        1    3.00
     6           2        1    3.00
     6           3        1    3.00
     7           2        1    3.00
     7           3        1    3.00
     8           2        1    3.00
     8           3        1    3.00
     9           2        1    3.00
     9           3        1    3.00
    10           2        1    3.00
    10           3        1    3.00
    11           2        1    3.00
    11           3        1    3.00
    12           2        1    3.00
    12           3        1    3.00
    13           2        1    3.00
    13           3        1    3.00
    14           2        1    3.00
    14           3        1    3.00
    15           2        1    3.00
    15           3        1    3.00
    16           2        1    3.00
    16           3        1    3.00
    17           2        1    3.00
    17           3        1    3.00
    18           2        1    3.00
    18           3        1    3.00
    19           2        1    3.00
    19           3        1    3.00
    20           2        1    3.00
    20           3        1    3.00
```

21	2	1	3.00
21	3	1	3.00
22	2	1	3.00
22	3	1	3.00
2	2	1	0.18
2	3	1	0.18
2	4	1	0.18
2	5	1	0.18
2	6	1	0.18
2	7	1	0.18
2	8	1	0.18
2	9	1	0.18
2	10	1	0.18
2	11	1	0.18
2	12	1	0.18
2	13	1	0.18
2	14	1	0.18
2	15	1	0.18
2	16	1	0.18
2	17	1	0.18
2	18	1	0.18
2	19	1	0.18
2	20	1	0.18
2	21	1	0.18
2	22	1	0.18
2	23	1	0.18
2	24	1	0.18
2	25	1	0.18
2	26	1	0.18
2	27	1	0.18
2	28	1	0.18
2	29	1	0.18
2	30	1	0.18
2	31	1	0.18
2	32	1	0.18
2	33	1	0.18
2	34	1	0.18
2	35	1	0.18
2	36	1	0.18
2	37	1	0.18
2	38	1	0.18
2	39	1	0.18
2	40	1	0.18
2	41	1	0.18
2	42	1	0.18
2	43	1	0.18
2	44	1	0.18
2	45	1	0.18
2	46	1	0.18

Table 4.--Input file for example application--Continued

2	47	1	0.18
2	48	1	0.18
2	49	1	0.18
2	50	1	0.18
2	51	1	0.18
2	52	1	0.18
3	12	1	0.18
3	13	1	0.18
3	14	1	0.18
3	15	1	0.18
3	16	1	0.18
3	17	1	0.18
3	18	1	0.18
3	19	1	0.18
3	20	1	0.18
3	21	1	0.18
3	22	1	0.18
3	23	1	0.18
3	24	1	0.18
3	25	1	0.18
3	26	1	0.18
3	27	1	0.18
3	28	1	0.18
3	29	1	0.18
3	30	1	0.18
3	31	1	0.18
3	32	1	0.18
3	33	1	0.18
3	34	1	0.18
3	35	1	0.18
3	36	1	0.18
3	37	1	0.18
3	38	1	0.18
3	39	1	0.18
3	40	1	0.18
3	41	1	0.18
3	42	1	0.18
3	43	1	0.18
3	44	1	0.18
3	45	1	0.18
3	46	1	0.18
3	47	1	0.18
3	48	1	0.18
3	49	1	0.18
3	50	1	0.18

3	51	1	0.18
3	52	1	0.18
4	29	1	0.18
4	30	1	0.18
4	31	1	0.18
4	32	1	0.18
4	33	1	0.18
4	34	1	0.18
4	35	1	0.18
4	36	1	0.18
4	37	1	0.18
4	38	1	0.18
4	39	1	0.18
4	40	1	0.18
4	41	1	0.18
4	42	1	0.18
4	43	1	0.18
4	44	1	0.18
4	45	1	0.18
4	46	1	0.18
4	47	1	0.18
4	48	1	0.18
4	49	1	0.18
4	50	1	0.18
4	51	1	0.18
4	52	1	0.18
5	33	1	0.18
5	34	1	0.18
5	35	1	0.18
5	36	1	0.18
5	37	1	0.18
5	38	1	0.18
5	39	1	0.18
5	40	1	0.18
5	41	1	0.18
5	42	1	0.18
5	43	1	0.18
5	44	1	0.18
5	45	1	0.18
5	46	1	0.18
5	47	1	0.18
5	48	1	0.18
5	49	1	0.18
5	50	1	0.18
5	51	1	0.18
5	52	1	0.18

The area is modeled with a grid of 53 nodes spaced 1.3 m horizontally by 36 nodes spaced 0.6 m vertically (table 4, data sets 5 and 6). There are five material types in the grid and each is listed in table 3 along with a description. The grid extends 10 nodes on the right beyond the pictured area in the cross section because sensitivity analyses results indicated that boundary effects from the artifical boundary were of the same order of magnitude as the modeled concentration when fewer nodes were used. Because the concentration gradients are expected to be larger in the z-direction due to the existence of horizontal layers of contrasting diffusivity, the nodes were spaced more closely in the z-direction than in the x-direction. Each layer is assumed to be isotropic with respect to diffusion. The data set describing the material properties distribution in the model grid is shown in table 4, data set 8.

The initial time step is set at 0.10 day, with a time-step multiplier of 1.5 to a maximum time step of 75 days. At the start of the simulation, the concentrations are changing rapidly, hence, the small initial time step. The convergence criterion was set at 0.001 Pa and the maximum number of iterations to 100. The time-step and convergence parameters were adjusted until mass balance error was less than 5 percent for all time steps. These parameters are shown in table 4, data sets 2 and 3.

The initial concentration (as a partial pressure) of methane is assumed to be atmospheric (0.18 Pa) everywhere. Concentration is input in units of pascals. The boundary conditions are constant throughout the simulation period of 15 years (table 4, data set 12). These are the concentration of methane at the trench and the concentration of methane in the atmosphere. The constant concentration nodes act as a source at the trench boundary and as a sink at the soil-atmosphere interface. These conditions are coded in table 4, data set 14.

A partial listing of the output file is shown in table 5. The table includes the entire output file to the end of the first time step and the concentrations at 5,447 days (time step 86). The concentration data at 5,447 days are contoured and plotted in figure 7.

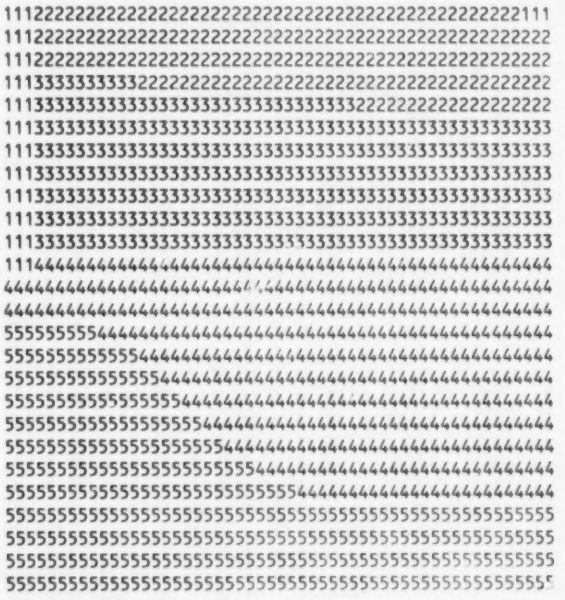
Diffusion to the atmosphere accounts for low concentration in the upper silt unit. The very low drained porosity of the clayey-silt unit effectively precludes diffusion through it. The methane in the sand unit moves laterally relatively quickly as escape to the atmosphere is cut off by the clayey silt above and the partially saturated, weathered shale bedrock unit below.

This application is a simple example of a field situation that demonstrates the effects of effective diffusivity and boundary conditions in two dimensions. The simulation of this problem on different computers (the IBM-AT and the SUN 3/260) produced slightly different results in the mass balance and number of iterations required. However, the concentrations were in excellent agreement.

Table 5.--Partial listing of output file for example application

```
DIFFUSION MODEL OUPUT
 .
123456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789

NUMBER OF STRESS PERIODS =         1
NUMBER OF COLUMNS =       53
NUMBER OF ROWS =       36
NUMBER OF DIFFERENT MATERIAL TYPES =         5
NUMBER OF PROPERTIES FOR EACH MATERIAL TYPE =         2
MAXIMUM NUMBER OF ITERATIONS PER TIME STEP =       100
NUMBER OF SPECIFIED PRINTOUT TIMES =         0

MAXIMUM SIMULATION TIME =    5600.00
INITIAL TIME STEP SIZE =     0.1000
TIME STEP MULTIPLIER =     1.5000
MAXIMUM TIME STEP SIZE =    75.0000
CLOSURE CRITERIA =     0.0010

DELX =

    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30
    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30
    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30
    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30
    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30    1.30
    1.30    1.30    1.30

DELZ =

    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60
    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60
    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60    0.60
    0.60    0.60    0.60    0.60    0.60    0.60

MATERIAL TYPE INDICES

11111111111111111111111111111111111111111111111111111
22222222111111111111111111111111111111111111111111111
22222222222111111111111111111111111111111111111111111
22222222222222222222222222221111111111111111111111111
11122222222222222222222222222222222222221111111111111
11122222222222222222222222222222222222221111111111111
11122222222222222222222222222222222222221111111111111
11122222222222222222222222222222222222221111111111111
11122222222222222222222222222222222222222111111111111
11122222222222222222222222222222222222222211111111111
```

Table 5.--Partial listing of output file for example application--Continued



DISTANCE (X OR R)

-6.500E-01	6.500E-01	1.950E+00	3.250E+00	4.550E+00	5.850E+00	7.150E+00	8.450E+00	9.750E+00	1.105E+01
1.235E+01	1.365E+01	1.495E+01	1.625E+01	1.755E+01	1.885E+01	2.015E+01	2.145E+01	2.275E+01	2.405E+01
2.535E+01	2.665E+01	2.795E+01	2.925E+01	3.055E+01	3.185E+01	3.315E+01	3.445E+01	3.575E+01	3.705E+01
3.835E+01	3.965E+01	4.095E+01	4.225E+01	4.355E+01	4.485E+01	4.615E+01	4.745E+01	4.875E+01	5.005E+01
5.135E+01	5.265E+01	5.395E+01	5.525E+01	5.655E+01	5.785E+01	5.915E+01	6.045E+01	6.175E+01	6.305E+01
6.435E+01	6.565E+01	6.695E+01							

Z	CONCENTRATIONS
-3.000E-01	1.800E-01 1.800E-01
3.000E-01	1.800E-01 1.800E-01
9.000E-01	1.800E-01 1.800E-01
1.500E+00	1.800E-01 1.800E-01
2.100E+00	1.800E-01 1.800E-01
2.700E+00	1.800E-01 1.800E-01

Table 5.--Partial listing of output file for example application--Continued

3.300E+00	1.800E-01 1.800E-01
3.900E+00	800E-01 1.800E-01
4.500E+00	1.800E-01 1.800E-01
5.100E+00	1.800E-01 1.800E-01
5.700E+00	1.800E-01 1.800E-01
6.300E+00	1.800E-01 1.800E-01

6.900E+00	1.800E-01 1.800E-01
7.500E+00	1.800E-01 1.800E-01
8.100E+00	1.800E-01 1.800E-01
8.700E+00	1.800E-01 1.800E-01
9.300E+00	1.800E-01 1.800E-01
9.900E+00	1.800E-01 1.800E-01

Table 5.--Partial listing of output file for example application--Continued

1.050E+01	1.800E-01 1.800E-01
1.110E+01	1.800E-01 1.800E-01
1.170E+01	1.800E-01 1.800E-01
1.230E+01	1.800E-01 1.800E-01
1.290E+01	1.800E-01 1.800E-01
1.350E+01	1.800E-01 1.800E-01

1.410E+01	1.800E-01 1.800E-01
1.470E+01	1.800E-01 1.800E-01
1.530E+01	1.800E-01 1.800E-01
1.590E+01	1.800E-01 1.800E-01
1.650E+01	1.800E-01 1.800E-01
1.710E+01	1.800E-01 1.800E-01

Table 5.--Partial listing of output file for example application--Continued

1.770E+01	1.800E-01 1.800E-01
1.830E+01	1.800E-01 1.800E-01
1.890E+01	1.800E-01 1.800E-01
1.950E+01	1.800E-01 1.800E-01
2.010E+01	1.800E-01 1.800E-01
2.070E+01	1.800E-01 1.800E-01

LENGTH OF NEW STRESS PERIOD = 5.600E+03 NUMBER OF NEW BOUNDARY CONDITIONS = 172

MATERIAL TYPE 1 1.862E+00 1.000E+00

EFFECTIVE DIFFUSION COEFFICIENT = 1.862E+00
EFFECTIVE STORAGE COEFFICIENT = 1.000E+00

MATERIAL TYPE 2 1.850E-01 1.000E+00

EFFECTIVE DIFFUSION COEFFICIENT = 1.850E-01
EFFECTIVE STORAGE COEFFICIENT = 1.000E+00

MATERIAL TYPE 3 4.012E-05 1.000E+00

EFFECTIVE DIFFUSION COEFFICIENT = 4.012E-05
EFFECTIVE STORAGE COEFFICIENT = 1.000E+00

MATERIAL TYPE 4 2.493E-01 1.000E+00

EFFECTIVE DIFFUSION COEFFICIENT = 2.493E-01
EFFECTIVE STORAGE COEFFICIENT = 1.000E+00

MATERIAL TYPE 5 1.700E-06 1.000E+00

EFFECTIVE DIFFUSION COEFFICIENT = 1.700E-06
EFFECTIVE STORAGE COEFFICIENT = 1.000E+00

RESULTS OF TIME STEP NUMBER 1
TIME = 0.1500
NUMBER OF ITERATIONS = 3
MAXIMUM HEAD CHANGE PER ITERATION =
0.4069 0.0025 0.0000

Z	CONCENTRATIONS
-3.000E-01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01

Table 5.--Partial listing of output file for example application--Continued

3.000E-01	1.800E-01 1.800E-01
9.000E-01	1.800E-01 2.018E-01 2.012E-01 1.809E-01 1.800E-01
1.500E+00	1.800E-01 5.062E-01 5.020E-01 1.894E-01 1.802E-01 1.800E-01
2.100E+00	1.800E-01 3.000E+00 3.000E+00 2.559E-01 1.811E-01 1.800E-01
2.700E+00	1.800E-01 3.000E+00 3.000E+00 2.602E-01 1.813E-01 1.800E-01
3.300E+00	1.800E-01 3.000E+00 3.000E+00 2.605E-01 1.813E-01 1.800E-01

3.900E+00	1.800E-01	3.000E+00	3.000E+00	2.605E-01	1.813E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
4.500E+00	1.800E-01	3.000E+00	3.000E+00	2.605E-01	1.813E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
5.100E+00	1.800E-01	3.000E+00	3.000E+00	2.605E-01	1.813E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
5.700E+00	1.800E-01	3.000E+00	3.000E+00	2.605E-01	1.813E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
6.300E+00	1.800E-01	3.000E+00	3.000E+00	2.605E-01	1.813E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
6.900E+00	1.800E-01	3.000E+00	3.000E+00	2.605E-01	1.813E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							

Table 5.--Partial listing of output file for example application--Continued

7.500E+00	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
8.100E+00	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
8.700E+00	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
9.300E+00	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
9.900E+00	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.050E+01	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							

1.110E+01	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.170E+01	1.800E-01	3.000E+00	3.000E+00	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.230E+01	1.800E-01	3.000E+00	3.000E+00	2.759E-01	1.819E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.290E+01	1.800E-01	5.844E-01	5.779E-01	1.952E-01	1.804E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.350E+01	1.800E-01	2.180E-01	2.168E-01	1.821E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.410E+01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							

Table 5.--Partial listing of output file for example application--Continued

1.470E+01	1.800E-01 1.800E-01
1.530E+01	1.800E-01 1.800E-01
1.590E+01	1.800E-01 1.800E-01
1.650E+01	1.800E-01 1.800E-01
1.710E+01	1.800E-01 1.800E-01
1.770E+01	1.800E-01 1.800E-01

1.830E+01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.890E+01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
1.950E+01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
2.010E+01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
2.070E+01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							

MASS BALANCE FOR TIME STEP NUMBER 1

		TOTAL	TIMESTEP
MASS THROUGH FIXED CONCENTRATION NODES	IN	0.18950E+01	0.18950E+01
	OUT	-0.26418E-02	-0.26418E-02
MASS THROUGH FIXED FLUX NODES	IN	0.00000E+00	0.00000E+00
	OUT	0.00000E+00	0.00000E+00
CHANGE IN STORAGE		0.18924E+01	0.18924E+01
MASS BALANCE		0.84657E-05	0.84657E-05
RELATIVE ERROR		0.26396E-07	0.26396E-07

DIFFUSION MODEL OUPUT

Table 5.--Partial listing of output file for example application--Continued

RESULTS OF TIME STEP NUMBER 86
TIME = 5446.7510
NUMBER OF ITERATIONS = 4
MAXIMUM HEAD CHANGE PER ITERATION =
0.0188 0.0095 0.0017 0.0005

Z	CONCENTRATIONS
-3.000E-01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01
3.000E-01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01
9.000E-01	1.800E-01 1.274E+00 1.209E+00 8.267E-01 5.801E-01 4.418E-01 3.588E-01 3.012E-01 2.392E-01 2.185E-01
	2.028E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01
1.500E+00	1.800E-01 2.382E+00 2.305E+00 1.445E+00 9.571E-01 6.919E-01 5.321E-01 4.239E-01 3.382E-01 2.876E-01
	2.499E-01 2.075E-01 1.969E-01 1.915E-01 1.881E-01 1.858E-01 1.842E-01 1.830E-01 1.822E-01 1.816E-01
	1.812E-01 1.809E-01 1.806E-01 1.805E-01 1.803E-01 1.803E-01 1.802E-01 1.801E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01
2.100E+00	1.800E-01 3.000E+00 3.000E+00 1.983E+00 1.287E+00 9.195E-01 6.943E-01 5.417E-01 4.297E-01 3.538E-01
	2.979E-01 2.506E-01 2.265E-01 2.120E-01 2.026E-01 1.961E-01 1.917E-01 1.885E-01 1.862E-01 1.845E-01
	1.833E-01 1.824E-01 1.818E-01 1.813E-01 1.810E-01 1.807E-01 1.805E-01 1.804E-01 1.802E-01 1.801E-01
	1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
	1.800E-01 1.800E-01 1.800E-01

2.700E+00	1.800E-01	3.000E+00	3.000E+00	2.275E+00	1.546E+00	1.117E+00	8.410E-01	6.509E-01	5.134E-01	4.159E-01
	3.440E-01	2.888E-01	2.541E-01	2.315E-01	2.164E-01	2.061E-01	1.989E-01	1.937E-01	1.900E-01	1.874E-01
	1.854E-01	1.840E-01	1.829E-01	1.821E-01	1.816E-01	1.812E-01	1.808E-01	1.806E-01	1.804E-01	1.802E-01
	1.802E-01	1.801E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
3.300E+00	1.800E-01	3.000E+00	3.000E+00	2.442E+00	1.742E+00	1.282E+00	9.695E-01	7.489E-01	5.887E-01	4.724E-01
	3.867E-01	3.227E-01	2.790E-01	2.494E-01	2.292E-01	2.154E-01	2.056E-01	1.987E-01	1.937E-01	1.900E-01
	1.874E-01	1.854E-01	1.840E-01	1.829E-01	1.821E-01	1.816E-01	1.811E-01	1.808E-01	1.806E-01	1.804E-01
	1.803E-01	1.802E-01	1.801E-01	1.801E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
3.900E+00	1.800E-01	3.000E+00	3.000E+00	2.542E+00	1.886E+00	1.415E+00	1.078E+00	8.341E-01	6.545E-01	5.225E-01
	4.247E-01	3.523E-01	3.010E-01	2.652E-01	2.407E-01	2.238E-01	2.118E-01	2.032E-01	1.970E-01	1.925E-01
	1.892E-01	1.867E-01	1.850E-01	1.836E-01	1.827E-01	1.820E-01	1.814E-01	1.810E-01	1.807E-01	1.805E-01
	1.803E-01	1.802E-01	1.802E-01	1.801E-01	1.801E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
4.500E+00	1.800E-01	3.000E+00	3.000E+00	2.605E+00	1.991E+00	1.520E+00	1.167E+00	9.055E-01	7.103E-01	5.652E-01
	4.573E-01	3.773E-01	3.196E-01	2.787E-01	2.506E-01	2.311E-01	2.173E-01	2.073E-01	2.000E-01	1.947E-01
	1.908E-01	1.879E-01	1.858E-01	1.843E-01	1.831E-01	1.823E-01	1.817E-01	1.812E-01	1.809E-01	1.806E-01
	1.804E-01	1.803E-01	1.802E-01	1.801E-01	1.801E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
5.100E+00	1.800E-01	3.000E+00	3.000E+00	2.644E+00	2.066E+00	1.599E+00	1.237E+00	9.627E-01	7.554E-01	6.000E-01
	4.839E-01	3.977E-01	3.347E-01	2.895E-01	2.587E-01	2.372E-01	2.219E-01	2.107E-01	2.026E-01	1.966E-01
	1.922E-01	1.890E-01	1.866E-01	1.849E-01	1.836E-01	1.826E-01	1.819E-01	1.814E-01	1.810E-01	1.807E-01
	1.805E-01	1.804E-01	1.802E-01	1.802E-01	1.801E-01	1.801E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							
5.700E+00	1.800E-01	3.000E+00	3.000E+00	2.670E+00	2.117E+00	1.656E+00	1.289E+00	1.005E+00	7.895E-01	6.264E-01
	5.042E-01	4.131E-01	3.459E-01	2.971E-01	2.648E-01	2.421E-01	2.256E-01	2.136E-01	2.047E-01	1.982E-01
	1.934E-01	1.398E-01	1.872E-01	1.853E-01	1.839E-01	1.829E-01	1.821E-01	1.815E-01	1.811E-01	1.808E-01
	1.806E-01	1.804E-01	1.803E-01	1.802E-01	1.801E-01	1.801E-01	1.801E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01	1.800E-01
	1.800E-01	1.800E-01	1.800E-01							

Table 5.--Partial listing of output file for example application--Continued

6.300E+00	1.800E-01 3.000E+00 3.000E+00 2.685E+00 2.149E+00 1.692E+00 1.323E+00 1.034E+00 8.122E-01 6.442E-01 5.178E-01 4.234E-01 3.532E-01 3.013E-01 2.688E-01 2.456E-01 2.284E-01 2.157E-01 2.063E-01 1.994E-01 1.942E-01 1.905E-01 1.877E-01 1.857E-01 1.842E-01 1.830E-01 1.822E-01 1.816E-01 1.812E-01 1.808E-01 1.806E-01 1.804E-01 1.803E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
6.900E+00	1.800E-01 3.000E+00 3.000E+00 2.692E+00 2.164E+00 1.710E+00 1.339E+00 1.048E+00 8.236E-01 6.531E-01 5.246E-01 4.285E-01 3.567E-01 3.013E-01 2.708E-01 2.478E-01 2.302E-01 2.171E-01 2.074E-01 2.002E-01 1.948E-01 1.909E-01 1.880E-01 1.859E-01 1.843E-01 1.832E-01 1.823E-01 1.817E-01 1.812E-01 1.809E-01 1.806E-01 1.804E-01 1.803E-01 1.802E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
7.500E+00	1.800E-01 3.000E+00 3.000E+00 1.890E+00 1.347E+00 1.067E+00 8.516E-01 6.824E-01 5.522E-01 4.533E-01 3.789E-01 3.236E-01 2.875E-01 2.960E-01 2.713E-01 2.488E-01 2.312E-01 2.178E-01 2.079E-01 2.006E-01 1.951E-01 1.911E-01 1.882E-01 1.860E-01 1.844E-01 1.832E-01 1.824E-01 1.817E-01 1.813E-01 1.809E-01 1.806E-01 1.805E-01 1.803E-01 1.802E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
8.100E+00	1.800E-01 3.000E+00 3.000E+00 1.071E+00 5.113E-01 4.113E-01 3.539E-01 3.098E-01 2.760E-01 2.503E-01 2.311E-01 2.170E-01 2.104E-01 2.438E-01 2.321E-01 2.194E-01 2.093E-01 2.016E-01 1.959E-01 1.917E-01 1.886E-01 1.863E-01 1.847E-01 1.834E-01 1.825E-01 1.818E-01 1.813E-01 1.810E-01 1.807E-01 1.805E-01 1.804E-01 1.803E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
8.700E+00	1.800E-01 3.000E+00 3.000E+00 8.376E-01 2.736E-01 2.249E-01 2.127E-01 2.043E-01 1.980E-01 1.932E-01 1.895E-01 1.869E-01 1.862E-01 1.958E-01 1.932E-01 1.900E-01 1.874E-01 1.855E-01 1.840E-01 1.830E-01 1.822E-01 1.816E-01 1.812E-01 1.809E-01 1.806E-01 1.805E-01 1.803E-01 1.802E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01
9.300E+00	1.800E-01 3.000E+00 3.000E+00 7.925E-01 2.277E-01 1.891E-01 1.858E-01 1.844E-01 1.834E-01 1.826E-01 1.820E-01 1.816E-01 1.815E-01 1.833E-01 1.828E-01 1.822E-01 1.817E-01 1.813E-01 1.810E-01 1.808E-01 1.806E-01 1.805E-01 1.804E-01 1.803E-01 1.802E-01 1.802E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01

9.900E+00	1.800E-01	3.000E+00	3.000E+00	7.930E-01	2.293E-01	1.916E-01	1.889E-01	1.879E-01	1.870E-01	1.862E-01
	1.855E-01	1.850E-01	1.845E-01	1.843E-01	1.838E-01	1.834E-01	1.830E-01	1.827E-01	1.824E-01	1.822E-01
	1.820E-01	1.818E-01	1.817E-01	1.815E-01	1.814E-01	1.813E-01	1.812E-01	1.811E-01	1.810E-01	1.809E-01
	1.809E-01	1.808E-01	1.807E-01	1.807E-01	1.806E-01	1.806E-01	1.806E-01	1.805E-01	1.805E-01	1.805E-01
	1.804E-01	1.804E-01	1.804E-01	1.804E-01	1.803E-01	1.803E-01	1.803E-01	1.803E-01	1.803E-01	1.803E-01
	1.803E-01	1.803E-01	1.800E-01							
1.050E+01	1.800E-01	3.000E+00	3.000E+00	8.417E-01	2.864E-01	2.450E-01	2.380E-01	2.327E-01	2.276E-01	2.229E-01
	2.189E-01	2.154E-01	2.120E-01	2.090E-01	2.063E-01	2.039E-01	2.018E-01	2.000E-01	1.984E-01	1.969E-01
	1.956E-01	1.944E-01	1.934E-01	1.924E-01	1.916E-01	1.908E-01	1.900E-01	1.894E-01	1.887E-01	1.882E-01
	1.877E-01	1.872E-01	1.867E-01	1.863E-01	1.860E-01	1.856E-01	1.853E-01	1.850E-01	1.847E-01	1.845E-01
	1.843E-01	1.841E-01	1.839E-01	1.837E-01	1.836E-01	1.835E-01	1.834E-01	1.833E-01	1.832E-01	1.832E-01
	1.831E-01	1.831E-01	1.800E-01							
1.110E+01	1.800E-01	3.000E+00	3.000E+00	1.094E+00	5.847E-01	5.261E-01	4.981E-01	4.723E-01	4.472E-01	4.236E-01
	4.037E-01	3.857E-01	3.686E-01	3.527E-01	3.387E-01	3.263E-01	3.153E-01	3.055E-01	2.967E-01	2.888E-01
	2.816E-01	2.752E-01	2.693E-01	2.639E-01	2.589E-01	2.544E-01	2.502E-01	2.462E-01	2.426E-01	2.393E-01
	2.362E-01	2.333E-01	2.306E-01	2.281E-01	2.258E-01	2.236E-01	2.216E-01	2.198E-01	2.180E-01	2.165E-01
	2.150E-01	2.137E-01	2.126E-01	2.115E-01	2.106E-01	2.098E-01	2.092E-01	2.086E-01	2.082E-01	2.079E-01
	2.077E-01	2.076E-01	1.800E-01							
1.170E+01	1.800E-01	3.000E+00	3.000E+00	1.992E+00	1.656E+00	1.550E+00	1.460E+00	1.372E+00	1.286E+00	1.204E+00
	1.135E+00	1.071E+00	1.011E+00	9.532E-01	9.025E-01	8.568E-01	8.162E-01	7.795E-01	7.463E-01	7.162E-01
	6.888E-01	6.636E-01	6.405E-01	6.191E-01	5.992E-01	5.807E-01	5.634E-01	5.473E-01	5.322E-01	5.180E-01
	5.048E-01	4.925E-01	4.809E-01	4.700E-01	4.597E-01	4.501E-01	4.411E-01	4.328E-01	4.250E-01	4.178E-01
	4.113E-01	4.053E-01	3.998E-01	3.950E-01	3.907E-01	3.870E-01	3.838E-01	3.812E-01	3.792E-01	3.777E-01
	3.767E-01	3.763E-01	1.800E-01							
1.230E+01	1.800E-01	3.000E+00	3.000E+00	2.890E+00	2.753E+00	2.623E+00	2.495E+00	2.368E+00	2.243E+00	2.122E+00
	2.020E+00	1.926E+00	1.835E+00	1.748E+00	1.671E+00	1.601E+00	1.538E+00	1.481E+00	1.429E+00	1.382E+00
	1.338E+00	1.298E+00	1.261E+00	1.226E+00	1.193E+00	1.163E+00	1.134E+00	1.107E+00	1.081E+00	1.057E+00
	1.034E+00	1.013E+00	9.927E-01	9.736E-01	9.555E-01	9.385E-01	9.224E-01	9.074E-01	8.934E-01	8.804E-01
	8.684E-01	8.574E-01	8.474E-01	8.385E-01	8.305E-01	8.236E-01	8.178E-01	8.129E-01	8.091E-01	8.064E-01
	8.045E-01	8.037E-01	1.800E-01							
1.290E+01	1.800E-01	2.995E+00	2.981E+00	2.879E+00	2.751E+00	2.623E+00	2.495E+00	2.368E+00	2.242E+00	2.118E+00
	2.019E+00	1.925E+00	1.834E+00	1.746E+00	1.669E+00	1.600E+00	1.537E+00	1.480E+00	1.429E+00	1.381E+00
	1.338E+00	1.298E+00	1.260E+00	1.226E+00	1.193E+00	1.163E+00	1.134E+00	1.107E+00	1.081E+00	1.057E+00
	1.034E+00	1.013E+00	9.928E-01	9.737E-01	9.556E-01	9.386E-01	9.226E-01	9.075E-01	8.935E-01	8.805E-01
	8.685E-01	8.575E-01	8.475E-01	8.386E-01	8.307E-01	8.238E-01	8.179E-01	8.131E-01	8.093E-01	8.065E-01
	8.047E-01	8.038E-01	1.800E-01							

Table 5.--Partial listing of output file for example application--Continued

1.350E+01	1.800E-01 2.990E+00 2.966E+00 2.873E+00 2.750E+00 2.623E+00 2.495E+00 2.368E+00 2.241E+00 2.110E+00 2.017E+00 1.925E+00 1.833E+00 1.742E+00 1.667E+00 1.597E+00 1.535E+00 1.479E+00 1.427E+00 1.380E+00 1.337E+00 1.297E+00 1.260E+00 1.225E+00 1.193E+00 1.162E+00 1.134E+00 1.107E+00 1.081E+00 1.057E+00 1.034E+00 1.013E+00 9.929E-01 9.738E-01 9.558E-01 9.387E-01 9.227E-01 9.077E-01 8.936E-01 8.806E-01 8.686E-01 8.576E-01 8.477E-01 8.387E-01 8.308E-01 8.239E-01 8.180E-01 8.132E-01 8.094E-01 8.066E-01 8.048E-01 8.040E-01 1.800E-01
1.410E+01	1.800E-01 3.185E-01 3.168E-01 3.105E-01 3.024E-01 2.941E-01 2.858E-01 2.778E-01 2.865E-01 2.093E+00 2.014E+00 1.924E+00 1.832E+00 1.734E+00 1.664E+00 1.593E+00 1.533E+00 1.476E+00 1.425E+00 1.378E+00 1.336E+00 1.296E+00 1.259E+00 1.225E+00 1.192E+00 1.162E+00 1.133E+00 1.106E+00 1.081E+00 1.057E+00 1.034E+00 1.013E+00 9.930E-01 9.739E-01 9.559E-01 9.388E-01 9.228E-01 9.078E-01 8.938E-01 8.807E-01 8.687E-01 8.577E-01 8.478E-01 8.388E-01 8.309E-01 8.240E-01 8.181E-01 8.133E-01 8.095E-01 8.068E-01 8.049E-01 8.041E-01 1.800E-01
1.470E+01	1.800E-01 1.818E-01 1.818E-01 1.817E-01 1.815E-01 1.814E-01 1.813E-01 1.812E-01 1.815E-01 2.604E-01 2.559E-01 2.505E-01 2.571E-01 1.721E+00 1.661E+00 1.586E+00 1.530E+00 1.472E+00 1.423E+00 1.376E+00 1.334E+00 1.294E+00 1.258E+00 1.224E+00 1.191E+00 1.161E+00 1.133E+00 1.106E+00 1.080E+00 1.057E+00 1.034E+00 1.013E+00 9.930E-01 9.740E-01 9.559E-01 9.389E-01 9.229E-01 9.079E-01 8.939E-01 8.808E-01 8.688E-01 8.579E-01 8.479E-01 8.389E-01 8.310E-01 8.241E-01 8.182E-01 8.134E-01 8.096E-01 8.069E-01 8.050E-01 8.042E-01 1.800E-01
1.530E+01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.809E-01 1.809E-01 1.808E-01 1.810E-01 2.385E-01 2.455E-01 1.575E+00 1.527E+00 1.466E+00 1.420E+00 1.372E+00 1.331E+00 1.292E+00 1.256E+00 1.223E+00 1.190E+00 1.160E+00 1.132E+00 1.105E+00 1.080E+00 1.056E+00 1.034E+00 1.013E+00 9.930E-01 9.740E-01 9.560E-01 9.390E-01 9.230E-01 9.080E-01 8.940E-01 8.810E-01 8.690E-01 8.580E-01 8.480E-01 8.390E-01 8.311E-01 8.242E-01 8.183E-01 8.135E-01 8.098E-01 8.070E-01 8.052E-01 8.043E-01 1.800E-01
1.590E+01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.806E-01 1.808E-01 2.302E-01 2.366E-01 1.458E+00 1.418E+00 1.367E+00 1.329E+00 1.289E+00 1.255E+00 1.221E+00 1.189E+00 1.159E+00 1.132E+00 1.105E+00 1.079E+00 1.056E+00 1.034E+00 1.013E+00 9.930E-01 9.741E-01 9.561E-01 9.391E-01 9.231E-01 9.081E-01 8.941E-01 8.811E-01 8.690E-01 8.581E-01 8.481E-01 8.391E-01 8.312E-01 8.243E-01 8.184E-01 8.136E-01 8.099E-01 8.071E-01 8.053E-01 8.044E-01 1.800E-01
1.650E+01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.805E-01 1.807E-01 2.236E-01 2.295E-01 1.360E+00 1.327E+00 1.285E+00 1.253E+00 1.220E+00 1.187E+00 1.158E+00 1.131E+00 1.104E+00 1.078E+00 1.055E+00 1.033E+00 1.013E+00 9.930E-01 9.741E-01 9.562E-01 9.392E-01 9.232E-01 9.082E-01 8.942E-01 8.811E-01 8.691E-01 8.581E-01 8.482E-01 8.392E-01 8.313E-01 8.244E-01 8.185E-01 8.137E-01 8.099E-01 8.072E-01 8.053E-01 8.045E-01 1.800E-01

1.710E+01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.804E-01 1.806E-01 2.184E-01 2.238E-01 1.279E+00 1.252E+00 1.219E+00 1.184E+00 1.157E+00 1.130E+00 1.103E+00 1.076E+00 1.054E+00 1.033E+00 1.013E+00 9.929E-01 9.741E-01 9.562E-01 9.392E-01 9.232E-01 9.082E-01 8.942E-01 8.812E-01 8.692E-01 3.582E-01 8.482E-01 8.393E-01 8.314E-01 8.245E-01 8.186E-01 8.138E-01 8.100E-01 8.073E-01 8.054E-01 8.045E-01 1.800E-01
1.770E+01	1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.800E-01 1.804E-01 1.805E-01 2.141E-01 2.128E-01 2.172E-01 1.179E+00 1.156E+00 1.130E+00 1.103E+00 1.074E+00 1.053E+00 1.033E+00 1.013E+00 9.929E-01 9.741E-01 9.562E-01 9.393E-01 9.233E-01 9.083E-01 8.943E-01 8.812E-01 8.692E-01 8.583E-01 8.483E-01 8.393E-01 8.314E-01 8.245E-01 8.186E-01 8.138E-01 8.101E-01 8.073E-01 8.054E-01 8.045E-01 1.800E-01
1.830E+01	1.800E-01 1.803E-01 1.803E-01 1.804E-01 2.091E-01 2.080E-01 2.068E-01 2.104E-01 1.070E+00 1.053E+00 1.033E+00 1.012E+00 9.929E-01 9.741E-01 9.562E-01 9.393E-01 9.233E-01 9.083E-01 8.943E-01 8.813E-01 8.693E-01 8.583E-01 8.483E-01 8.394E-01 8.314E-01 8.245E-01 8.186E-01 8.138E-01 8.101E-01 8.073E-01 8.054E-01 8.045E-01 1.800E-01
1.890E+01	1.800E-01 1.803E-01 1.803E-01 1.802E-01 1.803E-01 2.038E-01 2.031E-01 2.021E-01 2.012E-01 2.003E-01 1.995E-01 1.987E-01 1.980E-01 1.973E-01 1.966E-01 1.960E-01 1.955E-01 1.950E-01 1.945E-01 1.941E-01 1.938E-01 1.934E-01 1.932E-01 1.929E-01 1.927E-01 1.926E-01 1.925E-01 1.924E-01 1.923E-01 1.800E-01
1.950E+01	1.800E-01 1.802E-01 1.802E-01 1.802E-01 1.802E-01 1.802E-01 1.802E-01 1.802E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.801E-01 1.800E-01
2.010E+01	1.800E-01 1.800E-01
2.070E+01	1.800E-01 1.800E-01

Table 5.--Partial listing of output file for example application--Continued

MASS BALANCE FOR TIME STEP NUMBER 86

		TOTAL	TIMESTEP
MASS THROUGH FIXED CONCENTRATION NODES	IN	0.93763E+04	0.12721E+03
	OUT	-0.89478E+04	-0.12371E+03
MASS THROUGH FIXED FLUX NODES	IN	0.00000E+00	0.00000E+00
	OUT	0.00000E+00	0.00000E+00
CHANGE IN STORAGE		0.42511E+03	0.34613E+01
MASS BALANCE		0.34541E+01	0.37178E-01
RELATIVE ERROR		-0.33701E-01	0.11649E-03

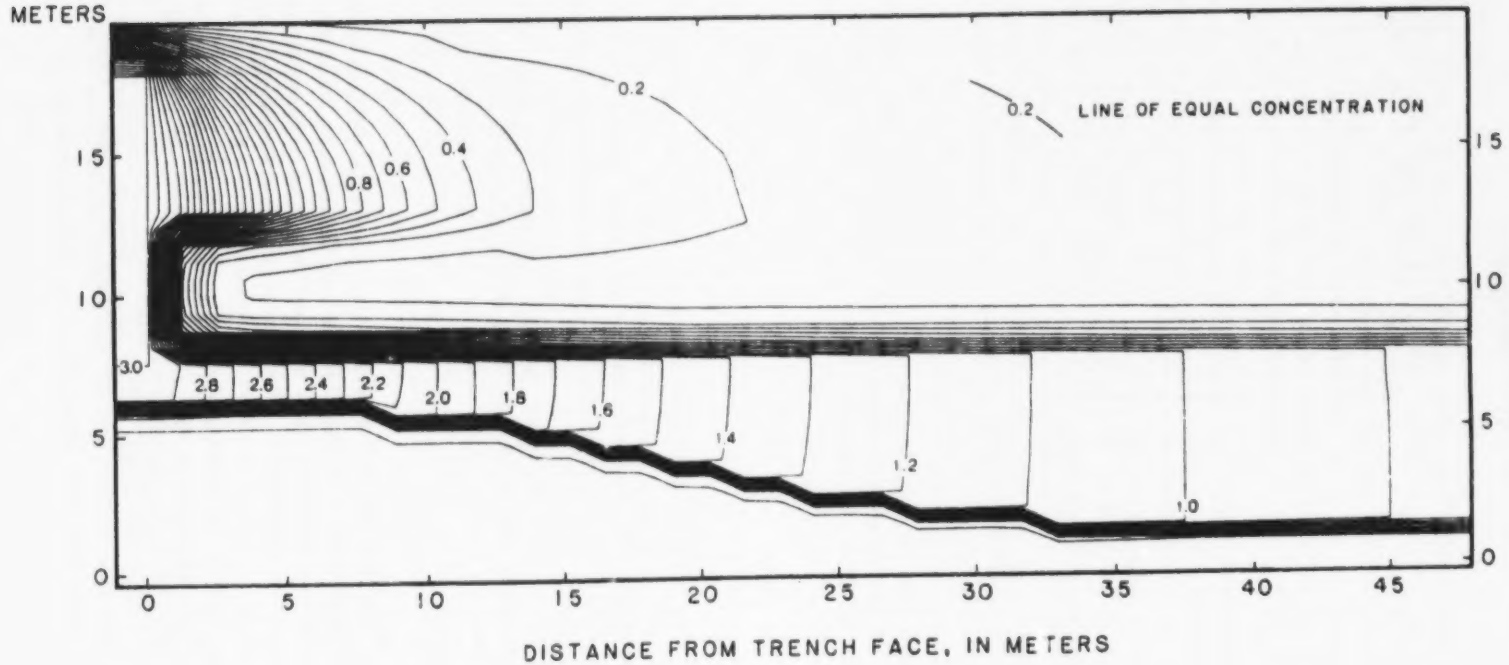


Figure 7.--Example of a plot of the program output.

SUMMARY

The diffusion equation describes transient flow for many physical phenomena. In hydrogeologic systems, examples include molecular diffusion, heat conduction, and fluid flow. A Fortran-77 computer code for the solution of the diffusion equation in one or two dimensions in Cartesian or cylindrical coordinates was presented in this report. The program may be applied to hydrogeologic systems in which the physical process is described by the diffusion equation. The flow media may be anisotropic and heterogeneous.

The equations describing these physical processes were briefly introduced. The derivation of the finite-difference approximations to produce matrix equations representing the processes for a block-centered grid was described. A wide range of problems may be simulated by changing the specifications for the initial and boundary conditions, and the descriptors of the conducting or diffusing media such as the anisotropy ratio and the constants of proportionality. The solution algorithm uses the method of preconditioned conjugate gradients, which does not require the estimation of iteration parameters and has desirable convergence properties. The mass balance is determined for each time step and for the total simulation.

Simulations were run to test the model results against analytical solutions for a variety of problems--diffusion in one dimension, heat transfer in an anisotropic media in two dimensions, and radial fluid flow in three dimensions using cylindrical coordinates. The results of the comparisons are presented in graphical form. There is good agreement for the cases tested, and mass balance for these tests was usually much less than 5 percent, indicating that the model functioned accurately and precisely under the tested conditions.

Application of the model to a hypothetical field situation of gas diffusion in the unsaturated zone was demonstrated. The grid design, determination of material parameters, and the selection of boundary conditions were discussed. The input and output files for the example application were included in tables.

The model is written in a modular format for ease of modification. Desirable modifications may include incorporation of specific formulations for the constants of proportionality that decouple the parameters describing the medium and the flux quantity, the addition of reaction terms linked to the flux quantity, and redimensioning of the arrays.

Simulations were run to test the model against analytical solutions for a variety of problems: molecular diffusion in one dimension, heat transfer in an anisotropic media in two dimensions, and radial fluid flow in three dimensions using cylindrical coordinates. The results of the comparisons are presented in graphical form. There is good agreement for the cases tested, and mass balance for these tests is generally much less than 5 percent, indicating that the model functions accurately and precisely under the tested conditions. Application of the model to a hypothetical field situation of gas diffusion in the unsaturated zone is demonstrated. The grid design, determination of material parameters, and the selection of boundary conditions are discussed. The input and output files for the example application are shown in tables.

REFERENCES CITED

Bird, R. B., Stewart, W. E., and Lightfoot, E. N., 1960, Transport phenomena: New York, John Wiley and Sons, 780 p.

Carslaw, H. S., and Jaeger, J. C., 1959, Conduction of heat in solids: Oxford, Oxford University Press, 510 p.

Concus, Paul, Golub, G. H., O'Leary D. P., 1976, A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations, in Bunch, James R., and Rose, D. J., eds., Sparse Matrix Computations: New York, Academic Press, p. 309-332.

Crank, John, 1956, The mathematics of diffusion: Oxford, Oxford University Press, 347 p.

Cunningham, R. E., and Williams, R. J. J., 1980, Diffusion in gases and porous media: New York, Plenum Press, 275 p.

Freeze, R. A., and Cherry, J. A., 1979, Groundwater: Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 604 p.

Katz, D. L., Cornell, David, Vary, J. A., Kobayashi, Riki, Elenbaas, J. R., Poettmann, F. H., Weinaug, C. F., 1959, Handbook of natural gas engineering: New York, McGraw-Hill Book Company, Inc., 802 p.

Konikow, L. F. and Bredehoeft, J. D., 1978, Computer model of two-dimensional solute transport and dispersion in ground water: U.S. Geological Survey Techniques of Water Resources Investigations, Book 7, Chapter C2, 90 p.

Kuiper, L. K., 1984, Documentation of a numerical code for the simulation of variable density ground-water flow in three dimensions: U.S. Geological Survey Water-Resources Investigations 84-4302, 24 p.

Lapidus, Leon, and Pinder, G. F., 1982, Numerical solution of partial differential equations in science and engineering: New York, John Wiley and Sons, 677 p.

Lohman, S. W., 1972, Ground-water hydraulics: U.S. Geological Survey Professional Paper 708, 70 p.

Meijerink, J.A., and van der Vorst, H. A., 1977, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix: Technical Report TR-1, Academic Computer Centre, Utrecht, the Netherlands, 29 p.

Mercer, J. W., and Faust, C. R., 1980, Ground-water modeling: Mathematical models: Ground Water, v. 18, p. 212-227.

Millington, R. J., and Quirk, J. M., 1960, Permeability of porous solids: Transactions of the Faraday Society, v. 57, p. 1200-1207.

Perry, R. H., Green, D. W., and Maloney, J. O., 1984, Perry's chemical engineers' handbook, sixth edition: New York, McGraw-Hill, various paging.

Remson, Irwin, Hornberger, G. M., and Molz, F. J., 1971, Numerical methods in subsurface hydrology: New York, Wiley-Interscience, 389 p.

Ritger, P. D., and Rose, N. J., 1968, Differential equations with applications: New York, McGraw-Hill, 545 p.

Theis, C. V., 1935, The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage: Transactions of the American Geophysical Union, v. 2, p. 519-524.

Trescott, P. C., Pinder, G. F., and Larson, S. P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, Chapter C1, 116 p.

Voss, C. I., 1984, A finite-element simulation model for saturated-unsaturated, fluid-density-dependent ground-water flow with energy transport or chemically-reactive single-species solute transport: U.S. Geological Survey Water-Resources Investigations Report 84-4369, 409 p.

Weeks, E. P., Earp, D. E., and Thompson, G. M., 1982, Use of atmospheric fluorocarbons F-11 and F-12 to determine the diffusion parameters of the unsaturated zone in the southern high plains of Texas: Water Resources Research, v. 18, p. 1365-1378.

Wong, Y. S., 1979, Preconditioned conjugate gradient methods for large sparse matrix problems, in Numerical methods in thermal problems, Lewis, R. W. and Morgan, K., eds.: Proceedings of the First International Conference, Pineridge Press, Swansea, United Kingdom, p. 967-979.

COMPUTER PROGRAM AND RELATED DATA

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES

The variables which appear in COMMON blocks are listed first, in the order in which they appear. The local variables are listed next for each subroutine in the order in which they appear. Common blocks are used to allocate more storage for static variables. Local variables do not use core storage. Hence, variables in common blocks have values available to all subroutines; local variable values must be passed to other subroutines. All arrays are real arrays unless noted otherwise. The constant of proportionality is referred to as diffusivity or the material parameter throughout.

Variable	Type and dimension	Description
	COMMON/RA	
CON	array 1936	Grid nodal concentration values
FLUX	array 1936	Grid nodal flux values
A	array 1936	DR/DEL - Interblock diffusivity in the z-direction divided by DEL which is computed recursively in the subroutine PRECON to do an incomplete decomposition of the matrix equation. matrix equation.
B	array 1936	DC/DEL - Interblock diffusivity in the x-direction divided by DEL which is computed recursively in the subroutine PRECON to do incomplete decomposition of the matrix equation.
D	array 20	Diffusivity for NMAT materials used in subroutine BOUND.
E	array 1936	Main diagonal matrix of the coefficient matrix, calculated in subroutine EQSETUP
RES	array 1936	Residual matrix, computed in EQSETUP.
DEL	array 1936	Computed in PRECON, used in CONGRAD to decompose the coefficient matrix.
DC	array 1936	Interblock diffusivity in the x-direction between the node indexed and the node to the left.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	COMMON/RA--Continued	
DR	array 1936	Interblock diffusivity in the z-direction between the node indexed and the node above.
COLD	array 1936	Grid nodal concentration values from previous time step, calculated in subroutine EQSETUP.
UR	array 1936	Intermediate matrix used to calculate the iteration parameters ALPH and BETA in subroutine CONGRAD, computed recursively.
PP	array 1936	Intermediate matrix used to calculate the iteration parameters ALPH and BETA in subroutine CONGRAD, the transpose of matrix UR.
AP	array 1936	Intermediate matrix used to calculate the iteration parameters ALPH and BETA in subroutine CONGRAD, the coefficient matrix multiplied by PP.
DELX	array 100	Distance between nodes in the x-direction, input in MAIN.
DELZ	array 100	Distance between nodes in the z-direction, input in MAIN.
PARAM	array 20x20	Parameter matrix for the material properties. Consists of diffusivity and up to 18 additional properties for each material. The last parameter is the anisotropy ratio. There are NPROP values required for each of NMAT material types.
PRINTT	array 20	Observation times for printouts, read in MAIN, only if NPER > 0.
TIME	real	Elapsed simulation time.

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	COMMON/RA--Continued	
CDELT	real	Multiplier for the time step size.
DELT	real	Time step size, input in MAIN.
RUR	array 1936	Intermediate matrix for calculating BETA in subroutine CONGRAD. It is equal to PP*RES.
P1	array 1936	Intermediate matrix for calculating the iteration parameter ALPH in subroutine CONGRAD. It is equal to AP*PP.
DELTMAX	real	Maximum time step size, input in MAIN.
TIMEMAX	real	Maximum simulation time input in MAIN.
BIGI	real	Closure error for each iteration, initialized to 0.0 in MAIN, set equal to TCHK in subroutine CONGRAD, returned to MAIN to compare with EPS to determine if another iteration is required.
S	array 20	Storage coefficient array. Calculated by calling function STOR1. Used as a divisor for the constant of proportionality to obtain an effective diffusivity or conductivity.
PI2	real	2 x π = 6.2834.
PLENGTH	real	Length of stress period to which boundary conditions apply.
R	array 22	Horizontal distance from the origin to the boundaries between nodes in the x-direction calculated in MAIN.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	COMMON/RA--Continued	
Z	array 22	Distance from the z surface to the boundaries between nodes in the z-direction, calculated in MAIN.
DHMX	array 200	Maximum change in concentration for each conjugate gradient iteration for the current time step.
NTYP	array 1936	Node types matrix, initialized in MAIN at 3 for interior nodes, -1 for boundary nodes. The node types may be changed to constant concentration or constant flux nodes in subroutine BOUND.
	COMMON/IA	
NMAT	integer	Number of material types (up to 20).
MAT	array 1936	Material types matrix.
NPR	integer	The number of specified output times.
NODES	integer	The number of nodes, determined as NCOL*NROW. Used throughout as an index stopping point.
NCOL	integer	Number of columns, input in MAIN.
NROW	integer	Number of rows, input in MAIN.
NSTEPS	integer	Current time step number. Initialized to 0 in MAIN and incremented in TIMER.
IT2	integer	Stopping indicator. Initialized to 0 in MAIN. If TIME exceeds PLENGTH or TIMEMAX, IT2 becomes -1 which ends simulation.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	COMMON/IA--Continued	
NPRR	integer	Index for times to print, initialized to 1 in MAIN, used in subroutine TIMER.
NPROP	integer	The number of parameters for each material in the media matrix. The minimum is 2: a constant of proportionality and the anisotropy ratio.
NMAT	integer	Number of different materials, up to 20.
IRAD	integer	Coordinate system indicator: 0 indicates Cartesian 1 indicates cylindrical.
MB	integer	The index identifier for the nodes at which flux is to be output to a separate file.
	COMMON/MASSB	
FLUX2	real	Total mass into fixed flux nodes.
CFIX2	real	Total mass into fixed concentration nodes.
CSTOR2	real	Total change in storage.
CFIX2A	real	Total mass out of fixed concentration nodes.
FLUX2A	real	Total mass out of fixed flux nodes.
CONIN	real	Total initial mass stored in the system.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	LOCAL VARIABLES	
	SUBROUTINE MAIN	
HEADER	character 80	Title and description of input file.
NPER	integer	Number of stress periods.
MAXIT	integer	Maximum number of iterations per time step.
IX	integer	Indicator for inputting DELX. If input as 0 then all values of array DELX are set equal to FACTX. If input as 1 then DELX is input and each value multiplied by FACTX.
FACTX	real	Multiplier or value for DELX.
IZ	integer	Indicator for inputting DELZ. If input as 0 then all values of array DELZ are set equal to FACTZ. If input as 1 then DELZ is input and each value is multiplied by FACTZ.
FACTZ	real	Multiplier or values for DELZ.
IC	integer	Indicator for inputting concentrations. If input as 0 then all values of CON equal FACTC. If input as 1 then all values of CON are multiplied by FACTC.
FACTC	real	Multiplier or value for CON.
NMB	integer	Number of nodes at which concentrations for intermediate time steps is to be written to a separate file.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	SUBROUTINE BOUND	
NB1	integer	Indicator for inputting material parameters. Input as 1 if new material parameters for the current stress period are to be read in, input as 0 otherwise.
NB2	integer	Number of nodes with new boundary conditions for the current stress period.
D1	real	Material property for the new stress period.
	SUBROUTINE TIMER	
D1	real	DELT * CDELT, size of new time step. Compared to DELTMAX to determine if the maximum time step size has been exceeded.
T1	real	Size of cumulative time steps. Used to check that total time has not exceeded TIMEMAX.
	SUBROUTINE EQSETUP	
K1,K2...	integers	Indices
N1,N2..N20	integers	Identifying numbers for material properties.
AREA	real	Cross-sectional area of block.
VOL	real	Volume of the block. If IRAD = 0, then VOL = DELX * DELZ; if IRAD = 1, then VOL = PI2 * DELX * DELZ * R.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	SUBROUTINE MASSB	
C	real array 4	Mass through fixed concentration nodes for a time step.
FF	array 20	Mass through fixed concentration nodes for selected nodes.
FLUX2	real	Total mass into fixed flux nodes.
CFIX2	real	Total mass into fixed concentration nodes.
FLUX1	real	Mass into fixed flux nodes for each time step.
CFIX1	real	Mass into fixed concentration nodes for each time step.
F	real	Holds the value of FLUX for a given node for computation.
FLUX1A	real	Mass out of fixed flux nodes for each time step.
CSTOR1	real	Change in storage for the nodes for the time step. Calculated as [CON(I) - COLD(I)] * S(N1) * VOL for each node, then summed for all nodes.
C1	array 4	Work array, set equal to C.
CFIX1	real	Mass into fixed concentration nodes for each time step.
CFIX1A	real	Mass out of fixed concentration nodes for each time step.
CMASS1	real	Mass balance for the time step.
CMASS2	real	Total mass balance.

ATTACHMENT A

DEFINITION OF PROGRAM VARIABLES--Continued

Variable	Type and dimension	Description
	SUBROUTINE MASSB--Continued	
REL1	real	Relative error for the time step. Equal to CMASS1/CSTOR1, which is mass residual divided by storage.
REL2	real	Relative error for the total time. Equal to CMASS2/STOR2, which is mass residual divided by storage.
FU	real	Mass through the upper boundary of a specified cell.
FD	real	Mass through the lower boundary of a specified cell.
FR	real	Mass through the right boundary of a specified cell.
FL	real	Mass through the left boundary of a specified cell.
FFF	real	Net mass change for the time step for a specified node written to file 10.

ATTACHMENT B

DATA INPUT REQUIREMENTS

Data set	Number of cards	Column	Format	Variable	Definition
1	1	1-80	A80	HEADER	Title to appear on output.
2	1	1-8	I8	NPER	Number of stress periods.
		9-16	I8	COL	Number of columns, up to 100.
		17-24	I8	ROW	Number of rows, up to 100.
		25-32	I8	NMAT	Number of different materials, up to 20.
		33-40	I8	NPROP	Number of properties for each material. Minimum is 2, maximum is 9.
		41-48	I8	MAXIT	Maximum number of iterations per time step.
		49-56	I8	NPR	Number of specified output times.
		57-64	I8	RAD	Coordinate system indicator, 0 indicates Cartesian coordinates, 1 indicates cylindrical coordinates.
		65-72	I8	NMB	Number of observation points for which results (flux and concentration) will be written to file 10 for each time step.
3	1	1-8	F8.0	TIMEMAX	Maximum simulation time.
		9-16	F8.0	DELT	Initial time step size.
		17-24	F8.0	CDELT	Multiplier for time step size.
		25-32	F8.0	DELTMAX	Maximum time step size.
		32-40	F8.0	EPS	Closure criterion.

ATTACHMENT B

DATA INPUT REQUIREMENTS--Continued

Data set	Number of cards	Column	Format	Variable	Definition
4	1	1-8	I8	IX	0 indicates grid spacing in x-direction is to be read for every column and multiplied by FACTX: 1 indicates constant grid spacing equal to FACTX.
		9-16	F8.0	FACTX	Factor or distance between nodes in x-direction.
		17-24	I8	IZ	0 indicates grid spacing in the z-direction is to be read in for every row and multiplied by FACTZ; 1 indicates constant grid spacing equal to FACTZ.
		25-32	F8.0	FACTZ	Factor or distance between nodes in z-direction
5	(NCOL/10)	1-80	10F8.0	DELX	Distance between nodes in the x-direction.
6	(NROW/10)	1-80	10F8.0	DELZ	Distance between nodes in the z-direction.
7	(NPR/10)	1-80	10F8.0	PRINTT	Observation times at which to print results. Present only if NPR > 0.
8	NROW	1-NCOL	NCOL*I1	MAT(I)	Material type of each node.
9	1	1-8	I8	IC	0 indicates constant factor for initial concentrations; 1 indicates the initial concentration values at all nodes equal FACTC.
		9-16	F8.0	FACTC	Constant factor for initial concentration or value of initial concentrations.

ATTACHMENT B

DATA INPUT REQUIREMENTS--Continued

Data set	Number of cards	Column	Format	Variable	Definition
10	NROW * (NCOL/10)	1-80	10F8.0	CON(I)	Initial values of concentration: each row must begin on a new card.
11	NMB	1-8	F8.0	I1	Observation node row number.
		9-16	F8.0	J1	Observation node column number. Present only if NMB > 0.
12	NPER	1-8	F8.0	PLENGTH	Length of stress period for following boundary conditions.
		9-16	I8	NB1	1 indicates new material parameter is present; 0 indicates none for new stress period.
		17-24	I8	NB2	Number of nodes with new boundary conditions.
13	NMAT	1-8	I8	K	Material type identifier.
		9-132	F12.0	PARAM	Material parameters. NPROP values required. The last value is the anisotoropy ratio, (constant of proportionality in the x-direction)/ (constant of proportionality in the z-direction)
14	NB2	1-8	I8	M	Row number of boundary node.
		9-16	I8	N	Column number of node.
		17-24	I8	NT	Type of node: 1 indicates Dirichlet; 2 indicates Neumann; 3 indicates interior node; -1 indicates exterior node;
		25-32	F8.0	C1	If NT=1, concentration at node; if NT=2, flux at node.

ATTACHMENT C

LIST OF PROGRAM ROUTINES

Figure 8 shows a generalized flow chart for the program execution. The specific names and functions of each program segment are listed in this attachment.

Subroutine MAIN

The overall execution of the program is controlled by subroutine MAIN. The model specifications are read including the grid dimensions, iteration parameters, time step parameters, and initial conditions. The arrays are initialized. The tests for convergence and maximum iterations exceeded are conducted in MAIN, and output of input data and program results except for mass balance are produced. All other subroutines are called from MAIN.

Subroutine BOUND

The length of time, boundary conditions and material properties are read in for each stress period. The functions for calculating the effective diffusivities and storativities are called. The results are written to the output file.

Function DIFF1

The function returns the first material parameter for the effective diffusivity unless the program is modified to calculate a value from additional material parameters.

Function STOR1

The effective storativity is returned as 1.0 unless the program is modified to calculate a value from the input parameters.

Subroutine TIMER

The current time step number and size is determined and checked against the limits input in MAIN.

Subroutine EQSETUP

The main diagonal and residual matrixes are calculated.

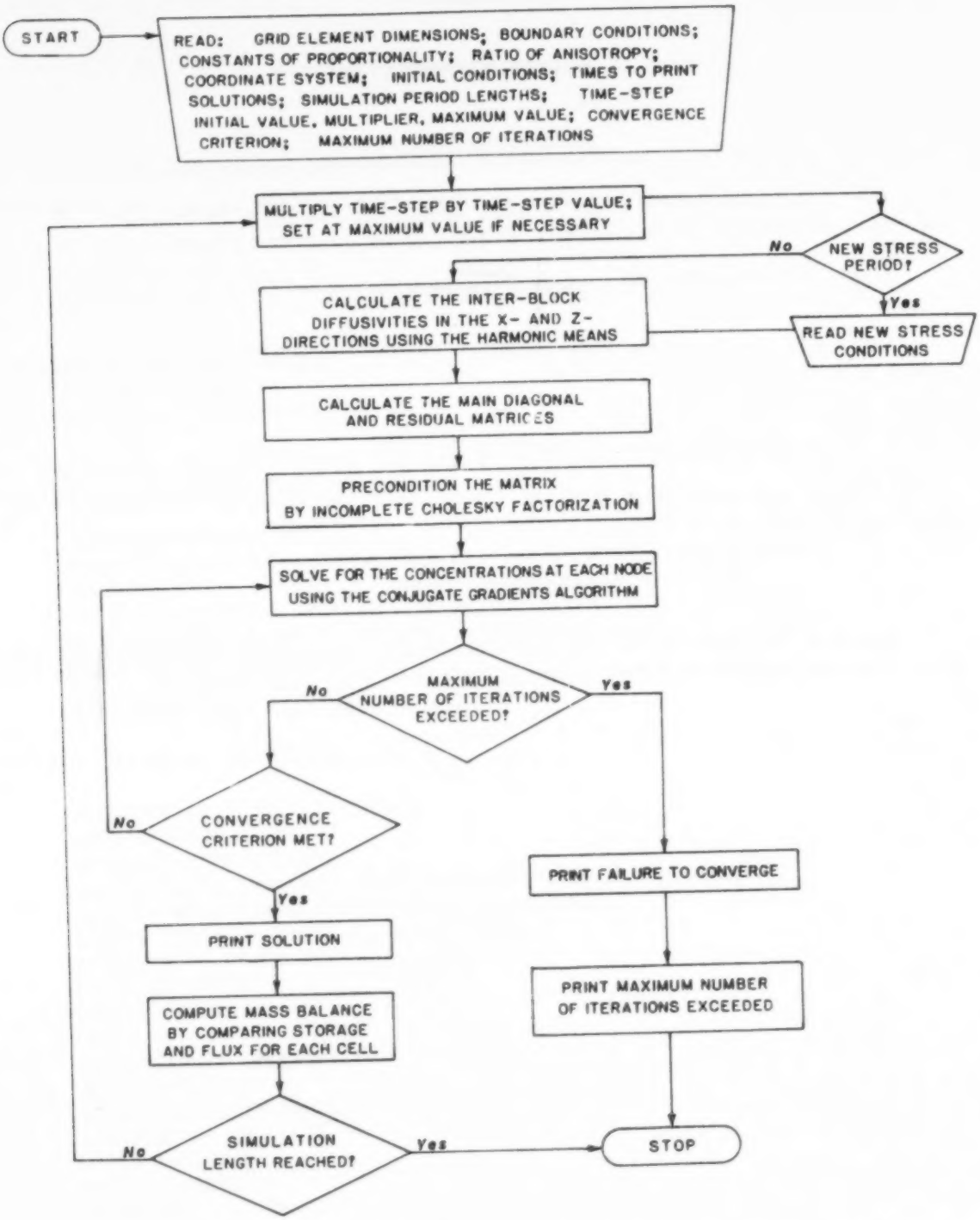

START
READ: GRID ELEMENT DIMENSIONS; BOUNDARY CONDITIONS; CONSTANTS OF PROPORTIONALITY; RATIO OF ANISOTROPY; COORDINATE SYSTEM; INITIAL CONDITIONS; TIMES TO PRINT SOLUTIONS; SIMULATION PERIOD LENGTHS; TIME-STEP INITIAL VALUE, MULTIPLIER, MAXIMUM VALUE; CONVERGENCE CRITERION; MAXIMUM NUMBER OF ITERATIONS
MULTIPLY TIME-STEP BY TIME-STEP VALUE; SET AT MAXIMUM VALUE IF NECESSARY
NEW STRESS PERIOD?
No
Yes
READ NEW STRESS CONDITIONS
CALCULATE THE INTER-BLOCK DIFFUSIVITIES IN THE X- AND Z-DIRECTIONS USING THE HARMONIC MEANS
CALCULATE THE MAIN DIAGONAL AND RESIDUAL MATRICES
PRECONDITION THE MATRIX BY INCOMPLETE CHOLESKY FACTORIZATION
SOLVE FOR THE CONCENTRATIONS AT EACH NODE USING THE CONJUGATE GRADIENTS ALGORITHM
MAXIMUM NUMBER OF ITERATIONS EXCEEDED?
No
Yes
CONVERGENCE CRITERION MET?
No
Yes
PRINT FAILURE TO CONVERGE
PRINT SOLUTION
PRINT MAXIMUM NUMBER OF ITERATIONS EXCEEDED
COMPUTE MASS BALANCE BY COMPARING STORAGE AND FLUX FOR EACH CELL
SIMULATION LENGTH REACHED?
No
Yes
STOP

Figure 8.--Simplified flow chart of major steps in program execution.

ATTACHMENT C

LIST OF PROGRAM ROUTINES--Continued

Subroutine CONDUCT

The interblock diffusivities (or conductivities) are calculated by using the harmonic mean of adjacent node values.

Subroutine PRECON

The coefficient matrix calculated in EQSETUP is preconditioned by an Incomplete Cholesky Factorization.

Subroutine CONGRAD

Performs one conjugate gradient iteration. The value of the residual is returned to MAIN to be compard to the closure criterion.

Subroutine MASSB

The mass residual is calculated and the relative error determined for the time step and the total simulation. The results are written to the output file.

ATTACHMENT D

PROGRAM LISTING

```
C*
C*    ****************************************************************
C*    *                                                              *
C*    *   PROGRAM TO SIMULATE DIFFUSION IN 1,2, OR 3 DIMENSIONS      *
C*    *        WRITTEN BY R.W. HEALY  3-87.                          *
C*    *                                                              *
C*    ****************************************************************
C*
C*
C****************************
      PROGRAM DIFFMOD
C****************************
C*
C*  READ MODEL SPECIFICATIONS, INITIALIZE VARIABLES, CONTROL FLOW,
C*  OUTPUT AND TERMINATION OF PROGRAM
C*
      CHARACTER*132 HEADER
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,PI,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
C*
C* READ AND WRITE INPUT DATA
C*
      PI2=6.283185
      READ(9,1000) HEADER
      WRITE(12,2000) HEADER
      READ(9,1001) NPER,NCOL,NROW,NMAT,NPROP,MAXIT,NPR,IRAD,NMB
      READ(9,1002) TIMEMAX,DELT,CDELT,DELTMAX,EPS
      READ(9,1005) IX,FACTX,IZ,FACTZ
      IF(IX.EQ.0) GO TO 10
      DO 11 I=1,NCOL
   11 DELX(I)=FACTX
      GO TO 13
   10 READ(9,1002) (DELX(I),I=1,NCOL)
      DO 12 I=1,NCOL
   12 DELX(I)=DELX(I)*FACTX
   13 IF(IZ.EQ.0) GO TO 17
      DO 15 I=1,NROW
   15 DELZ(I)=FACTZ
      GO TO 20
   17 READ(9,1002) (DELZ(I),I=1,NROW)
```

PROGRAM LISTING--Continued

```
      DO 18 I=1,NROW
   18 DELZ(I)=DELZ(I)*FACTZ
   20 R(1)=-.5*DELX(1)
      DO 201 J=2,NCOL
  201 R(J)=R(J-1)+0.5*(DELX(J)+DELX(J-1))                              50
      Z(1)=-.5*DELZ(1)
      DO 202 J=2,NROW
  202 Z(J)=Z(J-1)+0.5*(DELZ(J)+DELZ(J-1))
      WRITE(12,2001) NPER,NCOL,NROW,NMAT,NPROP,MAXIT,NPR
      IF(NPR.LE.0) GO TO 22                                            55
      READ(9,1002) (PRINTT(I),I=1,NPR)
      WRITE(12,2005) (PRINTT(I),I=1,NPR)
   22 PRINTT(NPR+1)=1E+22
      WRITE(12,2002) TIMEMAX,DELT,CDELT,DELTMAX,EPS
      WRITE(12,2003) (DELX(I),I=1,NCOL)                                60
      WRITE(12,2004) (DELZ(I),I=1,NROW)
      WRITE(12,2020)
      DO 23 J=1,NROW
      K=(J-1)*NCOL+1
      READ(9,1003) (MAT(K1),K1=K,K+NCOL-1)                             65
   23 WRITE(12,1003) (MAT(K1),K1=K,K+NCOL-1)
      READ(9,1005)IC,FACTC
      WRITE(12,2030) (R(I),I=1,NCOL)
      WRITE(12,2050)
      DO 24 J=1,NROW                                                   70
      K=(J-1)*NCOL+1
      K2=K+NCOL-1
      IF(IC.NE.0) GO TO 30
      READ(9,1006) (CON(K1),K1=K,K2)
      DO 28 K1=K,K2                                                    75
   28 CON(K1)=CON(K1)*FACTC
      ACON(K1)=CON(K1)
      GO TO 24
   30 DO 31 K1=K,K2
   31 CON(K1)=FACTC                                                    80
      ACON(K1)=CON(K1)
      WRITE(12,2060)
   24 WRITE(12,1004) Z(J),(CON(K1),K1=K,K2)
      IF(NMB.EQ.0) GO TO 240
      DO 242 I=1,NMB                                                   85
      READ(9,1001) I1,J1
  242 MB(I)=(I1-1)*NCOL+J1
  240 MB(NMB+1)=-1
```

PROGRAM LISTING--Continued

```
C*
C* INITIALIZE NODE TYPES
C*
      DO 25 J=1,NROW
      K=(J-1)*NCOL
      DO 25 I=1,NCOL
      K=K+1
      NTYP(K)=3
      FLUX(K)=0
      IF(J.EQ.1.OR.J.EQ.NROW.OR.I.EQ.1.OR.I.EQ.NCOL)NTYP(K)=-1
   25 CONTINUE
      IT2=0
      NSTEPS=0
      NPRR=1
      NODES=NCOL*NROW
      TIME=0.
C*
C*  BEGIN STRESS PERIODS LOOP.
C*
      DO 100 I8=1,NPER
C*
C*   RESET BOUNDARIES FOR NEW STRESS PERIOD.
C*
      CALL BOUND
C*
C*   CALCULATE INTERCELL CONDUCTANCES
C*
      CALL CONDUCT
C*
C*   DETERMINE TIME STEP SIZE.
C*
   50 CALL TIMER
C*
C*   SETUP FINITE DIFFERENCE EQUATIONS.
C*
      CALL EQSETUP
C*
C*   PERFORM INCOMPLETE DECOMPOSITION.
C*
      CALL PRECON
C*
C*   CONJUGATE GRADIENT LOOP
C*
      DO 55 I7=1,MAXIT
      CALL CONGRAD(I7)
```

```
      DHMX(I7)=BIGI                                                    135
      IF(BIGI.LT.EPS) GO TO 60
   55 CONTINUE
   60 WRITE(12,2010)NSTEPS,TIME,I7,(DHMX(K),K=1,I7)
      WRITE(12,2050)
      DO 65 I=1,NROW                                                   140
      K=(I-1)*NCOL+1
      WRITE(12,2060)
   65 WRITE(12,1004) Z(I),(CON(K1),K1=K,K+NCOL-1)
C*
C*    CALL MASS BALANCE ROUTINE                                        145
C*
      CALL MASSB
      IF(BIGI.LT.EPS) GO TO 70
      WRITE(12,2012)
      STOP                                                             150
   70 IF(IT2.EQ.0) GO TO 50
  100 CONTINUE
      WRITE(12,2013)
 1000 FORMAT(A80)
 1001 FORMAT(10I8)                                                     155
 1002 FORMAT(10F8.0)
 1003 FORMAT(100I1)
 1004 FORMAT(1PE10.3,'|',10(1PE10.3),9(/10X,'|',10(1PE10.3)))
 1005 FORMAT(I8,F8.0,I8,F8.0)
 1006 FORMAT(10F8.0)                                                   160
 2000 FORMAT('DIFFUSION MODEL OUPUT  ',//A132)
 2001 FORMAT(/,'NUMBER OF STRESS PERIODS = ',I8,
     &/,'NUMBER OF COLUMNS = ',I8,/,'NUMBER OF ROWS = ',I8,/,
     &'NUMBER OF DIFFERENT MATERIAL TYPES = ',I8,/,
     &'NUMBER OF PROPERTIES FOR EACH MATERIAL TYPE = ',I8,/,           165
     &'MAXIMUM NUMBER OF ITERATIONS PER TIME STEP = ',I8,/,
     &'NUMBER OF SPECIFIED PRINTOUT TIMES = ',I8)
 2002 FORMAT(/,'MAXIMUM SIMULATION TIME = ',F10.2,/,
     &'INITIAL TIME STEP SIZE = ',F10.4,/,
     &'TIME STEP MULTIPLIER = ',F10.4,/,                               170
     &'MAXIMUM TIME STEP SIZE = ',F10.4,/,
     &'CLOSURE CRITERIA = ',F10.4)
 2003 FORMAT(/,'DELX = ',/,10(/,10F8.2))
 2004 FORMAT(/,'DELZ = ',/,10(/,10F8.2))
 2005 FORMAT(/,'TIMES FOR PRINTOUTS = ',2(10F8.2))                     175
 2010 FORMAT(/,'RESULTS OF TIME STEP NUMBER ',I8,/,
     &'TIME = ',F12.4,/,'NUMBER OF ITERATIONS = ',I8,/,
     &'MAXIMUM HEAD CHANGE PER ITERATION = ',/,20(10F8.4,/))
 2012 FORMAT(/,'EXCEEDED MAXIMUM ITERATIONS')
```

PROGRAM LISTING--Continued

```
 2013 FORMAT('END OF SIMULATION')                                        180
 2020 FORMAT(/,'MATERIAL TYPE INDICES',/)
 2030 FORMAT(/,45X,' DISTANCE (X OR R)  ',/,
     &120('_'),/,12X,10(1PE10.3),9(/12X,10(1PE10.3)))
 2040 FORMAT(10('_'),'|',121('_'))                                       185
 2050 FORMAT(/,'     Z',40X,'  CONCENTRATIONS ')
 2060 FORMAT(120('_'))
      STOP
      END
C*                                                                       190
C*************************
      SUBROUTINE BOUND
C*************************
C*
C*   ROUTINE TO READ DATA FOR NEW STRESS PERIOD.                         195
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)                                 200
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
C*
C*    READ LENGTH OF STRESS PERIOD AND BOUNDARY CHANGES                  205
C*    NB1=1 SIGNIFIES THAT NEW SOIL PROPERTIES ARE TO BE READ
C*    NB2 IS THE NUMBER OF BOUNDARY CHANGES FROM PREVIOUS PERIOD
C*
      READ(9,1000) PLENGTH,NB1,NB2
      WRITE(12,2010) PLENGTH,NB2                                         210
      PLENGTH=TIME+PLENGTH
      IF(NB1.LT.1) GO TO 50
      DO 10 I=1,NMAT
      READ(9,1001) K,(PARAM(K,J),J=1,NPROP)
C*                                                                       215
C*  DETERMINE VALUES OF D AND S
C*
      D(K)=DIFF1(K)
      S(K)=STOR1(K)
      WRITE(12,2000) K,(PARAM(K,J),J=1,NPROP)                            220
   10 WRITE(12,2020) D(K),S(K)
   50 IF (NB2.LT.1) GO TO 70
C*
C*  READ IN NEW BOUNDARY CONDITIONS
C*
```

```
      DO 60 I=1,NB2                                                    225
      READ(9,1002) M,N,NT,C1
      N1=(M-1)*NCOL+N
      NTYP(N1)=NT
      IF(NT.EQ.1) CON(N1)=C1
      IF(NT.EQ.2) FLUX(N1)=C1                                          230
   60 CONTINUE
   70 CONTINUE
 1000 FORMAT(F8.0,2I8)
 1001 FORMAT(I8,10F12.0)
 1002 FORMAT(3I8,F8.0)                                                 235
 2000 FORMAT(/5X,'MATERIAL TYPE ',I2,2X,10(1PE10.3),2(/,22X,10(1PE10.3)
     &))
 2010 FORMAT(111('_'),//,5X,'LENGTH OF NEW STRESS PERIOD = ',
     &1PE10.3,10X,'NUMBER OF NEW BOUNDARY CONDITIONS = ',I5)
 2020 FORMAT(/,24X,'EFFECTIVE DIFFUSION COEFFICIENT = ',1PE10.3,       240
     &/,24X,'EFFECTIVE STORAGE COEFFICIENT = ',1PE10.3)
      RETURN
      END
C*
C*****************************                                       245
      FUNCTION DIFF1(K)
C*****************************
C*
C*  ROUTINE TO COMPUTE DIFFUSIVITY
C*                                                                    250
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)                              255
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
      DIFF1=PARAM(K,1)
      RETURN
      END                                                              260
C*
C*************************
      FUNCTION STOR1(K)
C*************************
C*                                                                    265
C*  ROUTINE TO COMPUTE STORAGE TERM
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
```

PROGRAM LISTING--Continued

```
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
      STOR1=1.0
      RETURN
      END
C*
C************************************
      SUBROUTINE TIMER
C************************************
C*
C*   ROUTINE TO DETERMINE TIME STEP SIZE
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936) B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
      NSTEPS=NSTEPS+1
      IT2=0
    5 D1=DELT*CDELT
      IF (D1.GT.DELTMAX) D1=DELTMAX
      T1=TIME+D1
      IF(T1.LT.TIMEMAX) GO TO 10
      T1=TIMEMAX
      IT2=-1
      GO TO 15
   10 IF(T1.LT.PLENGTH)GO TO 15
      T1=PLENGTH
      IT2=-1
   15 IF (T1.LT.PRINTT(NPRR)) GO TO 20
      T1=PRINTT(NPRR)
      NPRR=NPRR+1
      IT2=0
   20 CONTINUE
      DELT=T1-TIME
      TIME=T1
      RETURN
      END
```

```
C*
C*****************************                                          315
      SUBROUTINE CONDUCT
C*****************************
C*
C*    CALCULATE INTERCELL CONDUCTANCES
C*                                                                      320
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)                              325
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
      DO 100 J=1,NROW
      K=(J-1)*NCOL
      DO 100 I=1,NCOL                                                  330
C*
C*  COMPUTE INTERBLOCK DIFFUSIONAL CONDUCTANCES
C*
      K=K+1
      DC(K)=0                                                          335
      DR(K)=0
      IF(NTYP(K).LT.0) GO TO 100
      N1=MAT(K)
      K2=K-NCOL
      K3=K-1                                                           340
      N2=MAT(K2)
      N3=MAT(K3)
      AREA=DELX(I)
      IF(IRAD.EQ.1) AREA=AREA*PI2*R(I)
      D1=D(N1)/PARAM(N1,NPROP)                                         345
      D2=D(N2)/PARAM(N2,NPROP)
      IF(NTYP(K2).GE.0)DC(K)=2*AREA*D1*D2/(DELZ(J)*D2+
     &DELZ(J-1)*D1)
      AREA=DELZ(J)
      IF(IRAD.EQ.1) AREA=AREA*PI2*(R(I)-.5*DELX(I))                    350
      IF(NTYP(K3).GE.0)DR(K)=2*AREA*D(N1)*D(N3)/(DELX(I)*D(N3)+
     &DELX(I-1)*D(N1))
 100  CONTINUE
      RETURN
      END                                                              355
```

ATTACHMENT D

PROGRAM LISTING--Continued

```
C*
C*****************************
      SUBROUTINE EQSETUP
C*****************************                                          360
C*
C*    SET FINITE DIFFERENCE EQUATIONS
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),      365
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB                                    370
      DO 200 J=2,NROW-1
      DO 200 K=2,NCOL-1
      I=(J-1)*NCOL+K
C*
C*  COMPUTE MAIN DIAGONAL OF COEFFICIENT MATRIX AND RESIDUAL              375
C*
      E(I)=0
      RES(I)=0
      COLD(I)=CON(I)
      IF(NTYP(I).LT.0)GO TO 200                                          380
      N1=MAT(I)
      VOL=DELX(K)*DELZ(J)
      IF(IRAD.EQ.1) VOL=VOL*PI2*R(K)
      S1=VOL*S(N1)/DELT
      E(I)=-DC(I)-DC(I+NCOL)-DR(I)-DR(I+1)-S1
      RES(I)=-S1*COLD(I)-DC(I)*CON(I-NCOL)-DC(I+NCOL)*CON(I+             385
     &NCOL)-DR(I)*CON(I-1)-DR(I+1)*CON(I+1)-E(I)*CON(I)-FLUX(I)
 200  CONTINUE
      RETURN
      END                                                                390
C*
C*****************************
      SUBROUTINE PRECON
C*****************************
C*                                                                       395
C*    ROUTINE TO DO INCOMPLETE DECOMPOSITION OF MATRIX EQUATION
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,     400
```

```
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB
      DO 100 I=1,NODES
      A(I)=0                                                          405
      B(I)=0
      DEL(I)=0
      IF(NTYP(I).LT.0) GO TO 100
      IF(DEL(I-NCOL).EQ.0) GO TO 20
      A(I)=DC(I)/DEL(I-NCOL)                                          410
   20 IF(DEL(I-1).EQ.0) GO TO 30
      B(I)=DR(I)/DEL(I-1)
   30 DEL(I)=E(I)-B(I)*DR(I)-A(I)*DC(I)
  100 CONTINUE
      RETURN                                                          415
      END
C
C*********************************
      SUBROUTINE CONGRAD(NN1)
C*********************************                                   420
C*
C*   ROUTINE APLLIES ONE CONJUGATE GRADIENT ITERATION.
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),      425
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB                                430
      BIGI=0
      IF(NN1.NE.1) GO TO 10
      DO 8 I=1,NODES
      UR(I)=0
      PP(I)=0                                                         435
      AP(I)=0
      IF(NTYP(I).LT.2) GO TO 8
      UR(I)=RES(I)-B(I)*UR(I-1)-A(I)*UR(I-NCOL)
    8 CONTINUE
      R1=0                                                            440
      DO 9 I=NODES,1,-1
      IF(NTYP(I).LT.2) GO TO 9
      PP(I)=(UR(I)-PP(I+NCOL)*DC(I+NCOL)-PP(I+1)*DR(I+1))/DEL(I)
      R1=R1+PP(I)*RES(I)
    9 CONTINUE                                                        445
```

```
      RUR=R1
   10 P1=0
      DO 15 I=1,NODES
      IF(NTYP(I).LT.2) GO TO 15                                          450
      AP(I)=PP(I-NCOL)*DC(I)+PP(I+NCOL)*DC(I+NCOL)+PP(I-1)*DR(I)+
     1PP(I+1)*DR(I+1)+E(I)*PP(I)
      P1=P1+AP(I)*PP(I)
   15 CONTINUE
      ALPH=RUR/P1                                                        455
      DO 20 I=1,NODES
      IF(NTYP(I).LT.2) GO TO 20
      DELC=ALPH*PP(I)
      TCHK=ABS(DELC)
      IF(TCHK.GT.BIGI) BIGI=TCHK                                         460
      CON(I)=CON(I)+DELC
      RES(I)=RES(I)-ALPH*AP(I)
   20 CONTINUE
      DO 30 I=1,NODES
      UR(I)=0                                                            465
      IF(NTYP(I).LT.2) GO TO 30
      UR(I)=RES(I)-B(I)*UR(I-1)-A(I)*UR(I-NCOL)
   30 CONTINUE
      R1=0
      DO 40 I=NODES,1,-1                                                 470
      IF(NTYP(I).LT.2) GO TO 40
      UR(I)=(UR(I)-UR(I+NCOL)*DC(I+NCOL)-UR(I+1)*DR(I+1))/DEL(I)
      R1=R1+UR(I)*RES(I)
   40 CONTINUE
      BETA=R1/RUR                                                        475
      DO 50 I=1,NODES
      IF(NTYP(I).LT.2) GO TO 50
      PP(I)=UR(I)+BETA*PP(I)
   50 CONTINUE
      RUR=R1                                                             480
      RETURN
      END
C*
C***************************
      SUBROUTINE MASSB                                                   485
C***************************
C*
C*  ROUTINE TO COMPUTE MASS BALANCE AND OUTPUT RESULTS
C*
      COMMON/RA/CON(1936),FLUX(1936),A(1936),B(1936),D(20),E(1936),
     &RES(1936),DEL(1936),DC(1936),DR(1936),COLD(1936),UR(1936),        490
```

```
     &PP(1936),AP(1936),DELX(100),DELZ(100),PARAM(20,20),PRINTT(20),
     &TIME,CDELT,DELT,RUR,P1,DELTMAX,TIMEMAX,BIGI,S(20),PI2,PLENGTH,
     &R(100),Z(100),DHMX(200),ACON(1936)
      COMMON/IA/NTYP(1936),MAT(1936),NPR,NODES,NCOL,NROW,NSTEPS,IT2,
     &NPRR,NPROP,NMAT,IRAD,MB(100),NMB                                  495
      COMMON/MASS/FLUX2,CFIX2,CSTOR2,CFIX2A,FLUX2A,CFLUX1,CFLUX2,CONIN
      DIMENSION C(4),FF(20)
      SAVE FF
      IF(NSTEPS.GT.1) GO TO 10
      DO 5 I=1,20                                                       500
      FF(I)=0.0
    5 CONTINUE
      FLUX2=0
      FLUX2A=0
      CFIX2A=0                                                          505
      CFIX2=0
      CSTOR2=0
      DO 300 J=2,NROW-1
      DO 300 K=2,NCOL-1
      I=(J-1)*NCOL+K                                                    510
      VOL=DELX(K)*DELZ(J)
      IF(IRAD.EQ.1)VOL=VOL*PI2*R(K)
      CONIN=CONIN+(COLD(I)*VOL)
  300 CONTINUE
   10 FLUX1=0                                                           515
      CSTOR1=0
      FLUX1A=0
      CFIX1A=0
      CFIX1=0
      DO 200 J=2,NROW-1                                                 520
      DO 200 K=2,NCOL-1
      I=(J-1)*NCOL+K
      IF(NTYP(I).LT.0) GO TO 200
      IF(NTYP(I).EQ.1)GO TO 100
      IF(NTYP(I).NE.2) GO TO 50                                         525
      F=FLUX(I)
      IF(F.GE.0) THEN
      FLUX1=FLUX1+F
      ELSE
      FLUX1A=FLUX1A+F                                                   530
      END IF
   50 N1=MAT(I)
      VOL=DELX(K)*DELZ(J)
      IF(IRAD.EQ.1) VOL=VOL*PI2*R(K)
      CSTOR1=CSTOR1+(CON(I)-COLD(I))*S(N1)*VOL                          535
```

```
      GO TO 200
  100 N1=I-1
      N2=I+1
      N3=I-NCOL                                                          540
      N4=I+NCOL
      DO 102 NN=1,4
  102 C(NN)=0
      IF(NTYP(N1).GE.0)C(1)=+DR(I)*DELT*(CON(I)-CON(N1))
      IF(NTYP(N2).GE.0)C(2)=+DR(N2)*DELT*(CON(I)-CON(N2))                545
      IF(NTYP(N3).GE.0)C(3)=+DC(I)*DELT*(CON(I)-CON(N3))
      IF(NTYP(N4).GE.0)C(4)=+DC(N4)*DELT*(CON(I)-CON(N4))
      DO 150 I1=1,4
      C1=C(I1)
      IF(C1.GE.0) THEN                                                   550
      CFIX1=CFIX1+C1
      ELSE
      CFIX1A=CFIX1A+C1
      END IF
  150 CONTINUE                                                           555
  200 CONTINUE
      FLUX1=FLUX1*DELT
      FLUX1A=FLUX1A*DELT
      FLUX2=FLUX2+FLUX1
      CFIX2=CFIX2+CFIX1                                                  560
      FLUX2A=FLUX2A+FLUX1A
      CFIX2A=CFIX2A+CFIX1A
      CFLUX1=FLUX1+CFIX1+FLUX1A+CFIX1A
      CFLUX2=FLUX2+CFIX2+FLUX2A+CFIX2A
      CSTOR2=CSTOR2+CSTOR1                                               565
      CMASS2=FLUX2+CFIX2-CSTOR2+FLUX2A+CFIX2A
      CMASS1=FLUX1+CFIX1-CSTOR1+FLUX1A+CFIX1A
      IF(CSTOR2.EQ.0) THEN
      REL2=0
      ELSE                                                               570
      REL2=CMASS2/(CONIN-CSTOR2)
      END IF
      IF(CSTOR1.EQ.0) THEN
      REL1=0
      ELSE                                                               575
      REL1=CMASS1/(CONIN-CSTOR1)
      END IF
C*
C*  WRITE RESULTS
C*
      WRITE(12,1000)NSTEPS,CFIX2,CFIX1,CFIX2A,CFIX1A,FLUX2,FLUX1,FLUX2A 580
```

PROGRAM LISTING--Continued

```
     &FLUX1A,CSTOR2,CSTOR1,CMASS2,CMASS1,REL2,REL1
C*
C*   OUTPUT FLUXES AND CONC. AT SELECTED NODES TO FILE 10
C*
      DO 900 L=1,NMB                                                    585
      IF(MB(L).LT.0) GO TO 990
      L1=MB(L)
      L2=L1-NCOL
      L3=L1+NCOL
      L4=L1-1                                                           590
      L5=L1+1
      FU=DC(L1)*(CON(L1)-CON(L2))*DELT
      FD=DC(L3)*(CON(L1)-CON(L3))*DELT
      FL=DR(L5)*(CON(L1)-CON(L5))*DELT
      FR=DR(L1)*(CON(L1)-CON(L4))*DELT                                  595
      FFF=FU+FD+FR+FL
      FF(L)=FF(L)+FFF
      IF(L.EQ.1) WRITE(10,1050)
      WRITE(10,1100)L1,TIME,FFF,FF(L),CON(L1)
  900 CONTINUE                                                          600
  990 CONTINUE
 1000 FORMAT(//,'MASS BALANCE FOR TIME STEP NUMBER ',I3,/,T46,'TOTAL',
     &T60,'TIMESTEP',/,'MASS THROUGH FIXED CONCENTRATION NODES IN ',
     &T46,E14.5,T60,E14.5,/,'                                        OUT
     &',T46,                                                          605
     &E14.5,T60,E14.5,/,'MASS THROUGH FIXED FLUX NODES IN',T46,E14.5,
     &T60,E14.5,/,'                                  OUT',T46,E14.5,
     &T60,E14.5,/,'CHANGE IN STORAGE',T46,E14.5,T60,E14.5,/,
     &'MASS BALANCE',T46,E14.5,T60,E14.5,/,'RELATIVE ERROR',T46,
     &E14.5,T60,E14.5)                                                610
 1050 FORMAT(' NODE    TIME          MASS THROUGH FOR TIME STEP  TOTAL MA
     &SS THROUGH  CONCENTRATION')
 1100 FORMAT(I5,2X,E9.3,13X,E9.3,17X,E9.3,8X,E9.3)
      RETURN
      END                                                               615
```


☆U.S. GOVERNMENT PRINTING OFFICE: 1989 - 645-267